

```
{
  "id": "cnode:lenguaje-corp/code-structure.integration@1.0.0",
  "C": {
    "orchestrator": "Dada una raíz PATH, construye el árbol de código (estructura), extrae su firma estructural, la compara con otra(s) estructura(s) objetivo, negocia mappings y (opcionalmente) aplica traslado metafórico controlado para funciones/patrones; emite un plan de integración con adapters, umbrales y pruebas.",
    "nodes": [
      { "nombre": "path.scan", "rol": "Escanear PATH raíz y construir el árbol (dir→files→modules→deps)", "entrada": "PATH", "salida": "CodeTree" },
      { "nombre": "structure.signature", "rol": "Derivar firma: PARTS, RELS(contains/uses/depends), ORDER(secuencias), LAYERS", "entrada": "CodeTree", "salida": "StructureSignature(A)" },
      { "nombre": "target.load", "rol": "Cargar una o varias estructuras destino (paths o catálogos previos)", "entrada": "Targets[]", "salida": "StructureSignature(Bi)[]" },
      { "nombre": "structural.match", "rol": "Alinear A↔Bi y calcular métricas ( $\Omega_{\text{struct}}$ , C_layers,  $\Lambda_{\text{struct}}$ ,  $\kappa_{\text{order}}$ )", "entrada": "StructureSignature(A), StructureSignature(Bi)", "salida": "StructureMapping+Metrics[]" },
      { "nombre": "constraints.guard", "rol": "Verificar composition/adjacency/ordering y mínimos de cobertura", "entrada": "StructureMapping+Metrics[]", "salida": "OK | Violations[]" },
      { "nombre": "metaphor.option", "rol": "Si hay vecindad suficiente pero diferencias de vocab/roles, preparar traslado metafórico ( $\Omega$ , C,  $\Lambda$ ,  $\Phi$ )", "entrada": "Mappings válidos", "salida": "MetaphorSpec+Report[]" },
      { "nombre": "adapters.synth", "rol": "Generar adapters (type/edge alias, data mappers) y políticas de fusión", "entrada": "Mappings + (opcional) MetaphorReport", "salida": "AdaptersSpec" },
      { "nombre": "integration.plan", "rol": "Redactar plan de integración (pasos, riesgos, pruebas, rollback)", "entrada": "AdaptersSpec + Reports", "salida": "IntegrationPlan" },
      { "nombre": "emit", "rol": "Emitir paquete: mapping, adapters, métricas, pruebas sugeridas, provenance", "entrada": "IntegrationPlan", "salida": "IntegrationPackage" }
    ]
  }
}
```