```
{
  "id": "cnode:lenguaje-corp/structural-similarity.node-builder@1.0.0",
  "C": {
    "orchestrador": "Construye un Cognitive Node que modela 'similaridad estructural' entre A y B: extrae firmas de estructura (partes, relaciones, orden, capas), alinea firmas, calcula métricas y valida invariantes; emite un nodo reutilizable con mapping, métricas y provenance.",
    "nodos": [
      { "nombre": "intake.normalizer", "rol": "Canoniza A y B a grafos tipados", "entrada": "A,B heterogéneos", "salida": "Graph(A*), Graph(B*)" },
      { "nombre": "structure.extractor", "rol": "Deriva firmas estructurales", "entrada": "Graph(A*), Graph(B*)", "salida": "StructureSignature(A), StructureSignature(B)" },
      { "nombre": "signature.aligner", "rol": "Alinea partes/relaciones/orden", "entrada": "Firmas A,B", "salida": "StructureMapping {partsMap, relsMap, orderMap}" },
      { "nombre": "similarity.metrics", "rol": "Calcula Ω_struct, C_layers, Λ_struct, κ_order", "entrada": "Mapping + Firmas", "salida": "{OmegaStruct, CoverageLayers, LossinessStruct, KappaOrder}" },
      { "nombre": "constraints.guard", "rol": "Verifica composition/adjacency/ordering y mínimos", "entrada": "Métricas + Mapping", "salida": "OK | Violations[]" },
      { "nombre": "policy.decider", "rol": "Aplica umbrales y decide", "entrada": "Métricas + Violations", "salida": "{decision: ok|provisional|reject, recommendations[]}" },
      { "nombre": "node.emitter", "rol": "Construye el Cognitive Node final", "entrada": "Mapping + Métricas + Decision", "salida": "CognitiveNode<StructuralSimilarity>" }
    ]
  }
}
```