

ELEMENTO: "Análisis Estructural — pipeline canónico (general)"

alias: "analisis-estructural"

contexto: {

propósito: "Extraer la meta-information estructural de un artefacto (texto, código, audio, video, dataset, sistema) y emitir un Cognitive Node canónico reutilizable.",

forma_canónica: "{contexto, nodes, (metáforas|plantillas), evidencias, provenance}",

principios: [

"Parcimonia: el menor nº de pasos que cumpla el objetivo.",

"Compatibilidad formal: preservar composición/adyacencia/orden de la fuente.",

"Explicabilidad: toda afirmación debe trazar a evidencias.",

"Transferibilidad: destilar reglas/plantillas aplicables a otros dominios."

],

precondiciones: [

"Si input ≠ canónico → aplicar puente 'pre-procesar→estructura' (conexión inferida).",

"Definir parámetros de dominio: segmentación, ejes de balance, catálogo de macro-modelos."

],

parámetros: {

segmentación_fn: "cómo cortar el artefacto en partes mínimas",

ejes_balance: ["poder|influencia", "legitimidad|calidad", "información", "lealtad|cohesión"],

catálogo_macro: ["tres_actos", "arco_proceso", "pipeline_sistémico", "mosaico",

"jerarquía_capas"]

},

salidas: [

"tesis_global (string)",

"skeleton.grafo (macro/meso/micro)",

"balances[] por tramo",

"plantillas_transferibles[]",

"reporte.md / paquete CognitiveNode"

]

}

nodes: [

{ id: "N-0.intake",

rol: "Ingesta y pre-proceso",

acción: [

"Escanear/normalizar artefacto; construir manifest (parts crudas).",

"Si falta forma canónica → aplicar 'de-pre-procesar-a-estructura' (BridgeCraft)."

],

salida: ["artifact_struct", "provenance"]

},

{ id: "N0.observables",

rol: "Nivel 0 — Censo estructural",

acción: [

"Segregar en partes mínimas según segmentación_fn.",
"Registrar orden, adyacencias, composición y duración/weight.",
"Emitir parts[] y rels[] verificadas."
],
salida: ["parts[]", "rels[]"],
invariantes: ["cobertura≥95%", "orden consistente", "IDs únicos"],
requiere: ["N-0.intake"]
},

{ id: "N1.patrones_locales",
rol: "Nivel 1 — Motivos y micro-estructuras",
acción: [
"Detectar repeticiones/variaciones (motivos) y transiciones.",
"Mapear setup→payoff y distancia.",
"Anotar función por motivo (qué balance altera)."
],
salida: ["motifs[]", "payoffs[]", "transitions[]", "mapa_motivo→balance"],
requiere: ["N0.observables"]
},

{ id: "N2.dinámica",
rol: "Nivel 3 — Balances y umbrales",
acción: [
"Cuantificar ejes_balance por tramo (timeline).",
"Detectar umbrales/quiebres (cambios de estado)."
],
salida: ["tabla_balances[]", "umbrales[]"],
invariantes: ["clímax/turnos = quiebres en ≥2 ejes"],
requiere: ["N1.patrones_locales"]
},

{ id: "N3.arquitectura",
rol: "Nivel 2 — Macro-modelo y capas",
acción: [
"Probar catálogo_macro y justificar anclajes.",
"Construir skeleton.grafo por capas (macro/meso/micro).",
"Ubicar subtramas/módulos y su función sistémica."
],
salida: ["macroModel{}", "layers[]", "skeleton.grafo"],
requiere: ["N2.dinámica"]
},

{ id: "N4.thesis",
rol: "Abducción + falsación",
acción: [
"Proponer tesis candidatas (relación/flecha entre ejes).",
"Evaluar cobertura = escenas|módulos explicados / total.",
"Falsar con contraejemplos; elegir la mínima que explica el máximo."]

```

        ],
        salida: ["tesis_global", "pruebas_de_esfuerzo[]"],
        criterio_aceptación: "cobertura≈1 con menor costo explicativo",
        requiere: ["N3.arquitectura"]
    },
    {
        id: "N5.plantillas",
        rol: "Metáforas técnicas transferibles",
        acción: [
            "Condensar reglas en plantillas (cuándo_usar, pasos, señales_de_falla).",
            "Nombrar metáforas técnicas y, si aplica, diagrama ASCII."
        ],
        salida: ["plantillas_transferibles[]"],
        requiere: ["N4.tesis"]
    },
    {
        id: "N6.entrega",
        rol: "Redacción y publicación",
        acción: [
            "Redactar reporte.md con 10 apartados (visión, N0–N3, submódulos, tesis por componente si aplica, plantillas, conclusión).",
            "Empaquetar Cognitive Node canónico con provenance y métricas."
        ],
        salida: ["reporte.md", "CognitiveNodePackage"],
        requiere: ["N5.plantillas"]
    },
    {
        id: "N7.integración",
        rol: "Registro y reutilización",
        acción: [
            "Persistir en el grafo: {id, versión, interfaces}.",
            "Exponer API de consulta: parts(), skeleton(), balances(), plantillas(), tesis()."
        ],
        salida: ["graph_patch", "api_contract"],
        requiere: ["N6.entrega"]
    }
]

metáforas: [
    { nombre: "Trenza de submódulos", uso: "Sostener tensión/cohesión alternando hebras que mueven balances." },
    { nombre: "Ritual como compresor", uso: "Clímax donde forma visible promete X y dinámica muestra ¬X." },
    { nombre: "Legitimidad imposible", uso: "Cada paso a la luz exige sombra equivalente; cierre cobra la suma." }
]

```

```
métricas: {  
    cobertura_tesis: "0..1",  
    consistencia_orden: "violaciones por 1e3 edges",  
    fuerza_quiebre: " $\Delta$  en suma normalizada de ejes en umbrales",  
    transferibilidad: "#veces que una plantilla aplica fuera del artefacto"  
}
```

```
políticas: {  
    conexión_inferida: "si input≠canónico → aplicar 'pre-procesar→estructura' (bridge) con  
    trazabilidad",  
    rollback: "si falla un invariante, revertir a checkpoint anterior",  
    explicabilidad: "toda salida acompaña evidencias y rutas de cálculo"  
}
```