```
{
  "id": "cnode:lenguaje-corp/code.preprocessor@1.0.0",
  "C": {
    "orchestrator": "Convierte un árbol de código bajo ROOT_PATH en artefactos de texto tipados y trazables; genera un MANIFEST de estructura y contenidos normalizados, con chunking opcional para archivos grandes; preserva composición/adyacencia/orden para etapas posteriores.",
    "nodes": [
      { "nombre": "fs.scan", "rol": "Recorrer ROOT_PATH; recolectar metadatos por archivo", "entrada": "ROOT_PATH", "salida": "FileList[]" },
      { "nombre": "type.detect", "rol": "Detectar tipo/idioma/mediatype; marcar binarios y generados", "entrada": "FileList", "salida": "TypedFiles[]" },
      { "nombre": "transcribe.text", "rol": "Leer texto con encoding correcto; normalizar saltos de línea; NO tocar binarios", "entrada": "TypedFiles", "salida": "Transcribed[]" },
      { "nombre": "chunk.split", "rol": "Partir archivos grandes en chunks con offsets/lineMap", "entrada": "Transcribed", "salida": "Chunks[]" },
      { "nombre": "artifact.emit", "rol": "Emitir artefacto NDJSON por chunk/archivo con metadatos completos", "entrada": "Chunks", "salida": "Artifacts.ndjson" },
      { "nombre": "manifest.build", "rol": "Construir árbol (MANIFEST) con PARTS/RELS/ORDER; hashes y tamaños", "entrada": "FileList + Artifacts", "salida": "CodeManifest.json" },
      { "nombre": "provenance.record", "rol": "Registrar hash SHA256, mtime, toolVersion, filtros aplicados", "entrada": "Artifacts + Manifest", "salida": "ProvenanceTrace" },
      { "nombre": "emit", "rol": "Paquete reproducible para etapas siguientes", "entrada": "Artifacts + Manifest + ProvenanceTrace", "salida": "PreprocessedPackage" }
    ]
  }
}
```