Entropy | Odisea Cultura
Principal Software Engineer – Platform & Architecture
Technical Assessment

Overview:

Thank you for your interest in this role. This take-home assessment is designed to understand how you think about backend architecture, data processing, AI-assisted synthesis, and system design in the context of real-world construction project challenges.

This assessment uses a case study about the Northwest State University stadium renovation project. Your task is to design systems that can ingest and interpret real project artifacts (emails, Procore exports, daily reports, schedule updates, meeting minutes, etc.) and create a clear, defensible narrative of what happened during the project.

Estimated Effort: 2–4 hours (you may take more time if you wish)

Part 1 — Architecture Proposal:

Design the architecture of a platform that:

- Ingests data from sources such as Procore, email, PDFs, webcam frames, and meeting minutes.
- Extracts structured events (e.g., manpower issues, delays, design changes, weather impacts).
- Builds a chronological timeline of facts, events, and contradictions.
- Generates a clear story explaining what happened in the project.
- Preserves metadata, source lineage, and evidence provenance.

Please provide:

1. A high-level system architecture (diagram or narrative).
2. Description of the ingestion pipeline (email, Procore, PDFs, OCR, webcam, etc.).
3. How the system detects and handles missing, conflicting, or unreliable data.
4. How AI workflows (LLMs, embeddings, classification, summarization) support synthesis.
5. Proposed data model (entities and relationships).
6. Security considerations (multi-tenant, evidence chain-of-custody).
7. A 30/60/90-day roadmap.

Part 2 — Coding Exercise:

Please write a small NestJS service (real code or structured pseudocode) that:

1. Accepts a daily report PDF exported from Procore.
2. Stores the raw file and metadata (tenant, project, subcontractor).
3. Publishes an SQS message for processing.
4. Extracts structured elements such as weather, manpower, work areas, notes.
5. Stores extracted structured data with links back to the original evidence.
6. Includes at least one basic test.

Focus on clarity, modularity, and maintainability.

Part 3 — Written Responses:

Please provide short responses (3–6 sentences each):

1. How does your system handle contradictory narratives (e.g., disputed design change timing)?
2. If documentation is missing or inconsistent, how does your system reflect those gaps?
3. How would you prevent or mitigate AI hallucinations in a legal/claims context?
4. How would you challenge the founder respectfully if the system's findings contradict expectations?
5. Given limited time, what do you build first—and why?
