Prueba técnica — Senior Data Engineer (E2E)

Caso: WindyCity Cabs (empresa de taxis)

Objetivo

Diseñar e implementar un flujo **end-to-end** de datos: **API pública** → **ingesta incremental** → **procesamiento** → **modelo analítico** → **métricas** → **dashboards públicos**, mostrando tanto **capacidad de implementación** como **razonamiento de negocio** (qué medir, por qué, y cómo operarlo en producción).

Fuente de datos (API pública)

Usa un dataset público de viajes de taxi (Socrata / Chicago Data Portal).

Referencia del dataset (ID): `ajtu-isnz`

```
None
# Metadata (campos / tipos / descripción)
https://data.cityofchicago.org/api/views/ajtu-isnz

# Endpoint JSON (Socrata)
https://data.cityofchicago.org/resource/ajtu-isnz.json

# Más Info
https://data.cityofchicago.org/Transportation/Taxi-Trips-2024-/ajtu-isnz/about_data
```

> Alcance recomendado: **últimos 45–60 días** (elige tú y justifica). Si el volumen es alto, prioriza incrementalidad y datasets agregados para BI.

Timebox y expectativas

- **Timebox sugerido: 6–8 horas** (MVP).
- Priorizamos un entregable **ejecutable, claro y útil** por sobre “perfección”.
- Si no alcanzas todo, documenta en el README: **qué dejarías para después** y **por qué** (backlog + trade-offs).

Requisitos (Python obligatorio)

A) Ingesta incremental (raw + staging)

Implementa en **Python**:

1. Descarga desde la API con **paginación** y/o **partición por fecha**.
2. Guarda una capa **raw** (ej. JSON/Parquet) y una capa **staging** (limpia/tipada).
3. Define una estrategia de **incrementalidad** (watermark por timestamp, por ejemplo) que permita re-ejecutar sin duplicar ni romper (idempotencia práctica).

B) Transformación + modelo analítico (MVP consumible)

Crea un modelo mínimo para consumo analítico. Puedes elegir entre:

- **Opción 1 (star schema):** `fact_trips` + dimensiones (date/payment/company/areas).
- **Opción 2 (MVP recomendado):** `fact_trips` + **tablas agregadas** para BI (ej. `daily_kpis`, `hourly_kpis`, `zone_kpis`, etc.).

Incluye en README:

- granularidad (grain)
- llaves / claves
- campos derivados útiles (hour, weekday, etc.)
- supuestos y trade-offs

C) Métricas orientadas a negocio

Define primero **3–5 preguntas de negocio** que tu data product responderá (ej. revenue, eficiencia, mix de pago, zonas, estacionalidad, anomalías).

Luego define tus métricas (idealmente **8–12**). Para **cada métrica**:

- definición exacta

- cómo se calcula
- por qué importa (qué decisión habilita)

Dashboards públicos (obligatorio)

Debes entregar **4 dashboards públicos** (sin login) en una herramienta online.

Herramientas permitidas:

- Looker Studio
- Tableau Public
- O cualquier alternativa, **siempre que** el link sea **público y revisable** sin credenciales.

Dashboards requeridos (tú defines el contenido)

1. **Dirección/Finanzas — Dashboard #1**
2. **Dirección/Finanzas — Dashboard #2**
3. **Operación — Dashboard #1**
4. **Operación — Dashboard #2**

Importante: tú eliges qué mostrar en cada dashboard, pero debes incluir en el README:

- propósito del dashboard
- audiencia
- KPIs principales
- decisiones que habilita
- supuestos/limitaciones del dato

> Recomendación para el timebox: alimentar el BI con **tablas agregadas** (daily/hourly/zone) para hacerlo rápido, liviano y fácil de revisar.

Calidad de datos (MVP)

Implementa checks **mínimos**, automatizados y explícitos. Por ejemplo:

- claves/timestamps no nulos
- no negativos en montos/duración/distancia (si aplica)
- unicidad (trip_id o regla equivalente)
- manejo de duplicados
- outliers/anomalías: detecta y/o etiqueta (explica qué hiciste)

Uso de IA (permitido, con transparencia obligatoria)

Puedes usar herramientas de IA. Si las usas, debes incluir en el README:

- **Qué herramienta(s) usaste** (ej. ChatGPT, Copilot, Claude, etc.)
- **Qué prompts** utilizaste (copia y pega los prompts relevantes)
- **En qué partes** del trabajo la IA influyó (ej. diseño, código, SQL, ideas de dashboards, documentación)

Importante: si pasas a la siguiente fase, habrá una entrevista donde deberás **explicar y defender muchas de tus decisiones sin ayuda de IA**.

Entregables

1. **Link público al repositorio Git** (GitHub/GitLab) con:
    - código Python (ingesta + procesamiento)
    - SQL / transformaciones (si aplica)
    - README con:
        - cómo ejecutar end-to-end
        - decisiones y trade-offs
        - diagrama simple del pipeline (Mermaid/ASCII ok)
        - definiciones de métricas
        - qué datasets/tablas alimentan los dashboards
        - sección “Uso de IA” (si aplica)
        - backlog: qué harías después (si aplica)
2. **Links públicos** a los 4 dashboards (sin login)
3. (Opcional) 2–3 screenshots por dashboard (fallback si el link falla)

Restricciones técnicas

- **Lenguaje:** Python
- **BBDD**: Mysql
- Debe ser **reproducible** y con **instrucciones claras**
- Puedes usar herramientas/IA, siempre con la transparencia indicada arriba

---

Bonus (no obligatorio)

- Orquestación liviana (Makefile, task runner, Prefect/Airflow local)
- Tests automáticos + CI básico
- Observabilidad mínima (row counts, runtime, data freshness)
- Data dictionary más formal / catálogo
- Alertas simples por anomalías