# Prueba Técnica: Ingeniero de Datos Junior (Procesamiento de Datos)

## 1. Contexto del Desafío

Imagina que trabajas en una empresa de logística que monitorea su flota de vehículos. Cada vez que un vehículo inicia un viaje, genera un registro de evento. Tu equipo necesita analizar estos datos para tomar decisiones.

Tu tarea es procesar un archivo con los datos de viajes de un día y generar dos reportes clave.

## 2. Objetivo Principal

Escribir un script (en el lenguaje de tu preferencia) que lea un archivo de datos de origen y, como resultado, genere **dos nuevos archivos CSV** basados en los requisitos que se describen a continuación.

## 3. Fuente de Datos

Usaremos un conjunto de datos público de viajes de taxis de NYC. Para esta prueba, solo necesitas un archivo.

1. Descarga el archivo: Por favor, descarga un solo archivo Parquet del mes que prefieras (ej. "Yellow Taxi Trip Records") desde este enlace:  
   https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
2. **Campos Clave a Utilizar:**
   * tpep\_pickup\_datetime: Lo usaremos como el "timestamp" del evento.
   * PULocationID: Lo usaremos como el "ID del vehículo" (vehicle\_id).
   * pickup\_latitude: La latitud donde inició el viaje.
   * pickup\_longitude: La longitud donde inició el viaje.

*Nota: No te preocupes por los demás campos, puedes ignorarlos para esta prueba.*

## 4. Requisitos / Salidas Esperadas

Tu script debe leer el archivo Parquet de entrada y producir los siguientes dos archivos CSV de salida.

### Salida 1: "Última Ubicación Conocida"

Debes generar un archivo llamado ultima\_ubicacion.csv. Este archivo debe contener **solo una fila por cada vehicle\_id** (es decir, por cada PULocationID), mostrando la información del *último* viaje que realizó ese día.

El archivo debe tener las siguientes columnas:

* vehicle\_id
* ultimo\_timestamp (el tpep\_pickup\_datetime más reciente para ese vehículo)
* ultima\_latitud (la pickup\_latitude de ese último viaje)
* ultima\_longitud (la pickup\_longitude de ese último viaje)

**Lógica Clave:** Deberás encontrar el último registro para cada vehículo, basado en el tpep\_pickup\_datetime.

### Salida 2: "Reporte de Viajes por Hora"

Debes generar un archivo llamado viajes\_por\_hora.csv. Este archivo debe mostrar cuántos viajes *iniciaron* en cada hora del día (para todos los vehículos combinados).

El archivo debe tener las siguientes columnas:

* hora\_del\_dia (un número de 0 a 23)
* total\_viajes (el conteo total de viajes que iniciaron en esa hora)

**Lógica Clave:** Deberás extraer la "hora" del tpep\_pickup\_datetime, agrupar por esa hora, y contar el número de registros.

## 5. Herramientas Recomendadas

* **Lenguaje:** Te recomendamos usar **Python** y la librería **Pandas** (o **Polars**), ya que son herramientas muy comunes para este tipo de tareas y facilitan la lectura de Parquet y la manipulación de datos.
* **Flexibilidad:** Sin embargo, eres libre de usar cualquier lenguaje o herramienta con la que te sientas cómodo (ej. JavaScript/Node.js, Java, R, o incluso solo SQL si cargas los datos a una base de datos local como DuckDB).

## 6. Entregables

1. **Código Fuente:** Un enlace a un repositorio (Git) que contenga tu script.
2. **Instrucciones (README.md):** Un archivo README.md simple que explique:
   * Las herramientas que elegiste.
   * Cómo instalar cualquier dependencia (ej. pip install -r requirements.txt).
   * El comando exacto para ejecutar tu script (ej. python process\_data.py mi\_archivo.parquet).
3. **Archivos de Salida:** Por favor, incluye los archivos ultima\_ubicacion.csv y viajes\_por\_hora.csv generados en tu repositorio como muestra de que tu script funciona.
4. **Presentación Corta:** (Máx. 5 diapositivas) Una breve presentación (PDF o Google Slides) donde expliques tu enfoque:
   * ¿Qué herramientas elegiste y por qué?
   * ¿Qué desafíos encontraste?
   * ¿Cómo mejorarías tu solución si tuvieras más tiempo?
5. **Video Demo:** (Máx. 5 minutos) Un video corto (puedes subirlo a YouTube/Vimeo como "no listado" o a Google Drive) donde:
   * Muestres tu script en ejecución.
   * Expliques brevemente tu código.
   * Muestres los archivos de salida generados.

## 7. Criterios de Evaluación

* **Correctitud:** ¿Los archivos CSV de salida son correctos y cumplen con los requisitos?
* **Calidad del Código:** ¿El código es limpio, está bien organizado y es fácil de entender?
* **Claridad:** ¿Las instrucciones en el README.md son claras y permiten ejecutar tu script sin problemas?
* **Comunicación:** ¿La presentación y el video explican de forma clara tu solución?

*Nota Adicional: Entendemos que tienes otras ocupaciones. Estimamos que esta prueba puede tomar entre 2 y 4 horas de trabajo activo. Te recomendamos un plazo de entrega de 2 a 3 días para completarla con calma.*

¡Mucha suerte!