

Prueba Técnica Ingeniero de Datos Mid.

Eres **ingeniero(a) de datos en Jikkosoft** y te han encomendado liderar la **migración masiva** de información a un nuevo motor SQL. Para validar la viabilidad de la solución, debes construir una **Prueba de Concepto (PoC)** que logre lo siguiente:

1. Traslado del histórico

- Extraer los datos históricos almacenados en archivos CSV y cargarlos en la nueva base de datos.
- Por limitaciones de recursos, la carga de datos históricos deberá realizarse en lotes de hasta 1000 registros; el conjunto completo no debe mantenerse en memoria simultáneamente.

2. API REST para datos en línea

- Diseñar un servicio **REST** que reciba nuevas transacciones con estas características:
 - **Validación** contra el diccionario de datos antes de aceptar cada registro
 - Soporte para **lotes** de 1 - 1000 registros/filas por solicitud.
 - Capacidad de recibir información de **todas las tablas** a través del mismo endpoint.
 - Aplicación automática de las reglas de calidad específicas de cada tabla.

3. Respaldos en AVRO o PARQUET

- Implementar una función que genere copias de seguridad por tabla en formato **AVRO** o **PARQUET** y las almacene en un sistema de archivos.

4. Restauración

- Desarrollar el mecanismo que permita restaurar una tabla desde su respectivo backup **AVRO** o **PARQUET**.

Consideraciones

- Puedes decidir el origen donde se encontraran los archivos CSV.
- Elige la base de datos de destino, pero esta debe ser **SQL**.
- El archivo CSV está separado por comas.
- «Funcionalidad» puede interpretarse como servicio REST, procedimiento almacenado, funcionalidad propia de la base de datos, cron job u otra solución que cumpla los requisitos.
- Lenguajes sugeridos: Python, Go o Scala.

Extras que suman puntos

- Crear un archivo **README**.
- Incluir controles de **seguridad** para la API.
- Flujo de versionado GIT (branches, tags).
- Usar **Docker** (y Docker compose) para el empaquetado y uso de la APP.
- Uso de **servicios en la nube** en lugar de herramientas locales.
- Diagrama de la arquitectura propuesta.

- Incluye el uso de Kafka como veas conveniente en la solución.
 - Si incluyes Kafka, usa también Schema Registry.

Reglas de datos

- Las transacciones que no cumplan las reglas no deben insertarse; en su lugar, deben registrarse (log).
- Todos los campos son obligatorios

Estructura de los CSV

hired_employees.csv

Campo	Tipo	Descripción
id	INTEGER	Identificador del empleado
name	STRING	Nombre y apellido del empleado
datetime	STRING	Fecha/hora de contratación (ISO-8601)
department_id	INTEGER	Identificador del departamento contratado
job_id	INTEGER	Identificador del cargo asignado

departments.csv

Campo	Tipo	Descripción
id	INTEGER	Identificador del departamento
name	STRING	Nombre del departamento

jobs.csv

Campo	Tipo	Descripción
id	INTEGER	Identificador del cargo
name	STRING	Nombre del cargo

Desafío #2

Debes **analizar** los datos que fueron insertados en el **Desafío #1**.

Los **stakeholders** requieren ciertas métricas específicas, y deberás crear un endpoint independiente para cada solicitud.

Requisitos

- Cantidad de empleados contratados en X año por cargo y departamento, divididos por trimestre. La tabla debe ordenarse alfabéticamente por departamento y luego por cargo. Ejemplo de salida:

Department	Job	Q1	Q2	Q3	Q4
Staff	Recruiter	3	0	7	11
Staff	Manager	2	1	0	2
Supply Chain	Manager	0	1	3	0

- Lista de departamentos que contratan más empleados que el promedio de contrataciones en X año (considerando todos los departamentos).
 - Mostrar el ID, el nombre del departamento y la cantidad de empleados contratados
 - Ordenar los resultados de mayor a menor según la cantidad de contrataciones.

Ejemplo de salida:

ID	Department	Hired
1	Staff	12
7	Supply Chain	45

Extras que suman puntos

- Crea un informe visual para cada requerimiento utilizando tu herramienta favorita.

Entrega de solución

- Preferiblemente publica tu solución en **GitHub** o **GitLab**; se valorará un histórico de commits frecuentes que muestre la evolución del proyecto.
- Alternativamente, puedes entregar toda la solución en un archivo **.zip**.