EJERCICIO 1 - RESPUESTA AL INCISO

Para estructurar un proceso automatizado que permita crear y mantener un dataset confiable con los números de teléfono de los clientes, es necesario diseñar un pipeline de datos que ejecute esta tarea con la menor intervención manual posible. En este caso, sugeriría un pipeline batch que se ejecute con la periodicidad definida por reglas de negocio y que parta de fuentes de datos claramente identificadas. En terminos generales se podría decir que se debe construir primero un proceso ETL que permita a los datos ingresar a un flujo de trabajo en el que serán extraídos, verificados, transformados (si es necesario) y posteriormente cargados a un repositorio final al cual tendrán acceso las partes interesadas.

Antes de aplicar prácticas de CI/CD, el pipeline debe cumplir con la idempotencia, es decir, producir siempre el mismo resultado final al ejecutarse con las mismas entradas. Esto asegura consistencia y evita duplicaciones o errores acumulativos.

En el ciclo de CI/CD, se deben establecer entornos separados:

- Desarrollo (dev): espacio para implementar cambios y ejecutar pruebas iniciales.
- Staging: entorno que replica producción para validar la versión candidata antes del despliegue final.
- Producción (prod): entorno estable donde se ejecuta la versión validada del pipeline.

La aplicación de este enfoque permite a los equipos escribir código, integrarlo, ejecutar pruebas, entregar versiones, implementar cambios en el pipeline de forma colaborativa y en tiempo real, ejecutar automáticamente pruebas unitarias e integrales, realizar validaciones de calidad de datos, controles de seguridad y despliegues controlados con aprobaciones. Github actions es una herramienta que permite aplicar el proceso CI/CD de forma organizada y estructurada, además de permitir versionar el código a medida que se hagan cambios.

Finalmente, es recomendable incorporar mecanismos de observabilidad (monitoreo, alertas, métricas de calidad) y gobernanza (documentación de fuentes, transformaciones y propietarios del dataset), así como un sistema de versionado que permita conocer en todo momento qué versión del dataset está en uso.

En el repositorio he cargado un pipeline a manera de ejemplo en donde se detalle cada fase del proceso CI/CD con scripts ejemplo y además con las configuraciones necesarias en GitHub Actions que permitan observar más detalladamente el proceso CI/CD.

EJERCICIO 2 – RESPUESTA AL INCISO

Para poder observar la calidad de la información que está atravesando el pipeline sería necesario que en cada ejecución se calculen unas métricas clave que puedan crear un bosquejo de qué tan bien se encuentran los datos y si estos están dentro de los umbrales de calidad establecidos previamente por el negocio. Este proceso se llama observabilidad. La observabilidad hace referencia a medir y guardar de forma automática métricas y resultados de

checks en cada corrida del pipeline, para luego exponerlos como KPI’s de negocio que sean utilizables por las partes interesadas. En términos generales, podría decirse que esta información sobre la calidad de los datos está almacenada en una tabla de metadatos.

Además de la calidad, es importante incorporar la trazabilidad del dato, registrando el origen, transformaciones aplicadas, versión del pipeline y momento exacto de carga. Esto facilita auditorías y permite entender el historial de cada dato.

Algunos ejemplos de KPI’s que se podrían calcular durante cada ejecución del pipeline son: Porcentaje de duplicados, porcentaje de nulos, crecimiento neto de los datos (hoy vs ayer), frescura (días desde la ultima ejecución), etc. Estas métricas permiten conocer el estado de la información que está entrando al repositorio.

Esta información puede ser expuesta a los equipos de negocio a través de un dashboard o herramienta de visualización conectada al repositorio de metadatos, permitiendo así que los stakeholders monitoreen en tiempo real la calidad y el estado del dataset.