Diseño de flujo ETL batch para consolidación de clientes multi-origen

Contexto

Una empresa de retail posee sistemas independientes de registro de clientes en sus tiendas físicas, sitio web y aplicación móvil. Actualmente, la información está fragmentada, duplicada y sin un formato unificado. El área de analítica requiere contar con una base única, consolidada y actualizada diariamente para analizar comportamiento y fidelización.

Se propone un flujo de ingesta batch que permita consolidar registros de clientes desde múltiples fuentes, transformarlos y cargarlos a un repositorio centralizado.

1. Objetivos del flujo

- Consolidar información de clientes proveniente de tiendas físicas, sitio web y aplicación móvil.
- Eliminar duplicidades y normalizar datos para garantizar consistencia.
- Generar una base única diaria que permita análisis de comportamiento y estrategias de fidelización.
- Detectar y gestionar errores de forma proactiva mediante nodos de validación, logs y alertas.

2. Proceso ETL

a) Extracción

- **Orígenes de datos:**
    - Tiendas físicas: Base de datos SQL de punto de venta.
    - Sitio web: Exportaciones diarias en CSV o API REST.
    - Aplicación móvil: API REST con formato JSON.

- **Frecuencia:** Diaria, fuera del horario operativo (2:00 a.m. – 4:00 a.m.).
- **Formato de entrada:** SQL, CSV, JSON.

b) Transformación

- **Limpieza:** Eliminación de registros incompletos o con errores de formato.
- **Deduplicación:** Identificación de clientes duplicados usando combinación de campos clave (nombre, correo, teléfono).
- **Normalización:**
    - Estandarización de nombres y apellidos (capitalización correcta).
    - Normalización de correos electrónicos (minúsculas, eliminación de espacios).
    - Unificación de formatos de teléfonos y códigos de país.
- **Validación:**
    - Nodos que detecten valores nulos, formatos incorrectos o inconsistencias de tipo.
    - Rechazo de registros inválidos hacia un área de revisión manual.

c) Carga

- **Destino:** Base de datos centralizada (por ejemplo, en la nube: AWS RDS o Redshift).
- **Formato:** Tabla única de clientes con esquema normalizado y control de auditoría.

3. Herramienta ETL seleccionada

- **Apache NiFi**
  Justificación:
- Permite diseño visual de flujos de datos batch y streaming.
- Facilita integración con múltiples fuentes (CSV, JSON, SQL, APIs).

- Soporta nodos de validación, manejo de errores y rutas de datos alternativas.
- Registra automáticamente logs y permite configuración de alertas.

4. Buenas prácticas

- **Ventana de ejecución:** Programar la ingesta fuera del horario operativo (2:00 a.m. – 4:00 a.m.) para minimizar impacto en sistemas productivos.
- **Validación y control de errores:**
    - Crear nodos de verificación antes y después de cada transformación crítica.
    - Registrar errores en un log centralizado.
    - Configurar alertas por correo o Slack ante fallas en el flujo.
- **Monitoreo:** Mantener dashboards de ejecución diaria con indicadores: registros procesados, rechazados y duplicados.

5. Representación esquemática del flujo

Inicio del flujo – Programado a las 2:00 a.m.

1. **Extracción de datos**:
    - Conectar a SQL de tiendas físicas → exportar registros.
    - Consultar API del sitio web → descargar JSON o CSV.
    - Consultar API de app móvil → descargar JSON.
2. **Validación inicial de origen** – Detectar archivos vacíos o inconsistentes.
3. **Transformación de datos**:
    - Limpieza de registros incompletos.

    - Deduplicación de clientes por campos clave.
    - Normalización de nombres, correos y teléfonos.
4. **Validación final** – Comprobar consistencia de datos transformados.
5. **Carga en base centralizada** – Insertar registros limpios y unificados en la base de datos.
6. **Registro de logs y alertas** – Documentar errores y enviar notificaciones en caso de fallas.
7. **Fin del flujo** – Generación de reporte de ejecución diaria.