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

**1. Objetivos del flujo**

* **Problema a resolver**: Fragmentación y duplicación de registros de clientes en distintos sistemas (tiendas físicas, sitio web, app móvil).
* **Datos a consolidar**: Información personal (nombre, correo, teléfono), historial de compras, preferencias, y datos de fidelización.

**2. Proceso ETL**

**🔹 Extracción**

* **Orígenes**:
  + Archivos CSV exportados desde tiendas físicas.
  + API REST del sitio web.
  + Base de datos relacional de la app móvil.
* **Frecuencia**: Diaria, ejecutado durante la madrugada.
* **Formato de entrada**: CSV, JSON, SQL.

**🔹 Transformación**

* Limpieza de registros incompletos o inválidos.
* **Deduplicación** por correo electrónico o ID de cliente.
* Normalización de campos (nombres en mayúsculas, correos en minúsculas, formatos de fecha).
* Enriquecimiento con datos de fidelización y segmentación.

**🔹 Carga**

* **Destino**: Base de datos centralizada en la nube (Amazon RDS o Redshift).
* Organización por fecha de actualización y canal de origen.
* Generación de resumen de carga (cantidad de registros nuevos, actualizados, descartados).

**3. Herramienta ETL recomendada**

**Apache NiFi**

**Justificación**:

* Permite integrar múltiples fuentes (CSV, API, DB) de forma visual y modular.
* Soporta transformación de datos en tiempo de flujo.
* Incluye componentes para validación, manejo de errores y trazabilidad.
* Ideal para flujos programados y escalables sin necesidad de escribir código.

**4. Buenas prácticas**

* **Ventana de ejecución**: Programar el flujo entre 2:00 y 4:00 AM para evitar interferencias con sistemas operativos.
* **Validación de datos**: Nodos que detectan registros incompletos o duplicados antes de la carga.
* **Manejo de errores**:
  + Archivos fallidos se redirigen a una carpeta de cuarentena.
  + Logs detallados por cada etapa del flujo.
  + Alertas automáticas por correo en caso de fallas críticas.

5. Esquema técnico del flujo

[1] Recepción de datos desde:

     - CSV (tiendas físicas)

     - API (sitio web)

     - DB (app móvil)

     ↓

[2] Extracción con Apache NiFi:

     - ListFile / InvokeHTTP / QueryDatabaseTable

     ↓

[3] Transformación:

     - CleanRecord (limpieza)

     - DeduplicateRecord (deduplicación)

     - UpdateAttribute (normalización)

     ↓

[4] Validación:

     - RouteOnAttribute (registros válidos vs inválidos)

     ↓

[5] Carga:

     - PutDatabaseRecord (base centralizada)

     ↓

[6] Logs y alertas:

     - LogAttribute / PutEmail / PutFile (errores)