Propuesta de Arquitectura de Integración de Datos

Fecha: 2025-09-22

Empresa: FastTrack Logistics

**Resumen ejecutivo**

Diseño una propuesta de integración de datos para FastTrack Logistics que centralice las fuentes operacionales (ERP, WMS, TMS, sensores IoT, CRM, e‑commerce, carriers externos) en un repositorio unificado. La solución combina ingestión por lotes y streaming, transformación gobernada, y un catálogo con trazabilidad, priorizando escalabilidad, seguridad y calidad de datos para habilitar dashboards operacionales y analítica avanzada.

# 1. Diagnóstico del Entorno Actual

FastTrack Logistics cuenta con múltiples sistemas operativos que gestionan distintas áreas del negocio:

- ERP: Sistema de planificación de recursos empresariales que requiere integración batch para consolidar datos periódicos.

- Sensores IoT: Dispositivos que generan datos en tiempo real, ideales para integración vía streaming.

- CRM: Sistema de gestión de relaciones con clientes que expone datos mediante APIs REST.

- Bases de datos locales y hojas de cálculo utilizadas por equipos operativos.

Problemas y desafíos clave:

* Datos fragmentados en silos (formatos y frecuencias diferentes).
* Falta de consistencia e identificación única de entidades (clientes, pedidos y SKUs).
* Eventos de IoT no aprovechados en near‑real time.
* Ausencia de catálogo/linaje y políticas de calidad.
* Requerimientos de seguridad y auditoría para datos sensibles.

# 2. Arquitectura Propuesta

Se propone una arquitectura híbrida basada en herramientas open source y servicios cloud para integrar y centralizar los datos en un repositorio único:

- Apache NiFi para orquestación de flujos de datos.

- Talend para procesos ETL batch desde ERP y bases de datos.

- Apache Kafka para ingestión de datos en tiempo real desde sensores IoT.

- AWS Glue para transformación y catalogación de datos en la nube.

- Amazon S3 como repositorio centralizado de datos.

- Denodo para virtualización de datos y acceso unificado.

# 3. Herramientas Seleccionadas y Justificación

- Apache NiFi: Permite diseñar flujos de integración visuales y manejar múltiples formatos.

- Talend: Potente herramienta ETL con conectores para ERP y bases de datos.

- Kafka: Ideal para manejar grandes volúmenes de datos en tiempo real.

- AWS Glue: Servicio serverless que facilita la transformación y catalogación de datos.

- Denodo: Permite acceder a datos distribuidos sin necesidad de moverlos físicamente.

# 4. Estrategia de Calidad y Seguridad de los Datos

# Ingestión

# Apache NiFi (open‑source): orquestación de flujos, transformaciones ligeras, manejo de protocolos (SFTP, HTTP, Kafka). Excelente para equipos que necesitan trazabilidad y control visual.

# Apache Kafka (con Kafka Connect): backbone de mensajería para eventos e IoT, tolerante a fallos, permite desacoplar productores/consumidores y soporta stream processing.

# Debezium (CDC) o AWS DMS si se usa AWS: captura de cambios desde bases de datos para sincronizar con el Lakehouse.

# Ingestión y Procesamiento streaming

# Apache Flink o Kafka Streams: procesamiento de eventos con baja latencia (agregaciones, enriquecimiento de eventos, detección de anomalías).

# Almacenamiento

# Object Storage: S3 (AWS) / Azure Blob / GCS o MinIO on‑prem.

# Delta Lake o Apache Iceberg: transacciones ACID sobre objetos, versionado, time travel — ideal para patrones ETL/ELT reproducibles.

# Orquestación y ETL/ELT

# Apache Airflow: orquestación de pipelines batch/ELT, programación y retries.

# DBT (data build tool): transformaciones SQL versionadas en el layer transform (mantenimiento y tests de calidad).

# Data Warehouse / Serving

# Snowflake / Amazon Redshift Serverless / Databricks + Delta: para consultas ad‑hoc, BI y dashboards. Elegir según presupuesto y preferencia cloud.

# Catálogo, Linaje y Observabilidad

# Amundsen o DataHub: catálogo de datos y discovery.

# OpenLineage / Marquez: captura de linaje entre jobs.

# Prometheus + Grafana para métricas infra.

# Seguridad y CI/CD

# Vault para secrets management.

# KMS (Cloud native) para cifrado de datos en reposo.

# IAM / RBAC y políticas de red (VPC, Security Groups).

# Terraform para infra como código y reproducibilidad.

# Por qué esta combinación

# Permite cubrir ingestion patterns mixtos (batch + streaming).

# Usa herramientas maduras y ampliamente adoptadas con comunidad activa.

# Facilita gobernanza, trazabilidad y reproducibilidad (DBT + Delta + Airflow + Catalog).

# 5. Ejemplo de Flujo de Integración

Fuente: Sensores IoT → Kafka → NiFi → AWS Glue → S3 → Denodo → Dashboard

1. Kafka recibe datos en tiempo real desde sensores.

2. NiFi enruta los datos hacia AWS Glue.

3. Glue transforma y cataloga los datos.

4. Los datos se almacenan en S3.

5. Denodo accede a los datos en S3 y los expone a dashboards.

# 6. Beneficios Esperados

- Centralización de datos operativos y estratégicos.

- Mejora en la toma de decisiones basada en datos en tiempo real.

- Reducción de errores y duplicados.

- Mayor trazabilidad y seguridad en el acceso a la información.

