IT Academy
by KIBERNUM

Automatización de Flujos de Datos con Apache NiFi y Otras Plataformas

por Kibernum Capacitación S.A.

¿Qué es una plataforma de automatización de flujos de datos?

Una plataforma de automatización de flujos de datos es una herramienta que permite gestionar, mover y transformar datos de manera eficiente y en tiempo real entre diferentes sistemas o aplicaciones. Estas plataformas facilitan la integración de datos, la automatización de procesos y el manejo de grandes volúmenes de información que fluyen constantemente en sistemas empresariales o plataformas de TI.

Apache NiFi es un ejemplo destacado de plataforma de automatización de flujos de datos. Permite la ingesta, el procesamiento y el transporte de datos de una forma visual y sencilla mediante una interfaz gráfica, sin necesidad de escribir mucho código. NiFi es altamente escalable y flexible, y se usa comúnmente en proyectos de integración de datos y en sistemas que requieren mover grandes volúmenes de información de forma eficiente.

Objetivos y beneficios de la automatización de flujos de datos

La automatización de flujos de datos busca mejorar la eficiencia operativa, reducir errores y facilitar la toma de decisiones basada en datos en tiempo real. A continuación, se detallan los objetivos principales de la automatización de flujos de datos y los beneficios asociados.

Optimizar la Gestión de Datos

Permite manejar grandes volúmenes de información de manera eficiente, reduciendo la carga operativa.

Mejorar la Velocidad de Procesamiento

Acelera el procesamiento de datos para obtener información valiosa en menor tiempo.

Garantizar la Calidad y Consistencia

Asegura que los datos mantengan su integridad durante todo el proceso.

Reducir la Intervención Manual

Minimiza la necesidad de intervención humana en procesos repetitivos, disminuyendo errores.

Facilitar la Escalabilidad

Permite adaptar los sistemas a volúmenes crecientes de datos sin comprometer el rendimiento.

Integrar Diversas Fuentes de Datos

Facilita la conexión entre diferentes sistemas y formatos de datos.

Componentes y etapas de un flujo de datos

Un flujo de datos es el movimiento continuo de datos entre diferentes sistemas, aplicaciones o servicios. En el contexto de plataformas como Apache NiFi o Apache Kafka, el flujo de datos involucra varios componentes y etapas para asegurar que los datos se transmitan, procesen, y gestionen de manera eficiente. A continuación, se detallan los principales componentes y las etapas de un flujo de datos.

1

Productores (Sources)

Son las fuentes que generan o envían los datos hacia un sistema de procesamiento de flujos de datos. Estas pueden ser aplicaciones, bases de datos, sensores IoT, servicios web, archivos o cualquier otro sistema que produzca datos.

2

Procesadores (Transformadores)

Los procesadores son los componentes que manipulan los datos a medida que se mueven a través del flujo. Estos pueden realizar tareas como transformación, filtrado, validación, enriquecimiento o agregación de los datos.

3

Consumidores (Sinks)

Son los destinos donde los datos procesados son almacenados o utilizados. Los consumidores pueden ser bases de datos, almacenamiento en la nube, sistemas de análisis o aplicaciones que consumen los datos procesados.

4

Canales de Comunicación

Son los medios a través de los cuales los datos se transmiten entre los productores, procesadores y consumidores. Pueden ser colas de mensajes, bases de datos intermedias o servicios de transmisión de datos.

5

Almacenamiento de Datos (Opcional)

En algunos flujos de datos, los datos deben ser almacenados temporalmente antes de ser procesados o después de ser procesados. El almacenamiento puede ser en forma de bases de datos, sistemas de archivos o almacenamiento en la nube.

6

Sistemas de Monitoreo y Gestión

Son herramientas utilizadas para monitorear el rendimiento del flujo de datos, asegurarse de que no haya fallos en el proceso y gestionar el flujo en tiempo real.

Plataformas populares para automatizar flujos de datos

Apache Airflow

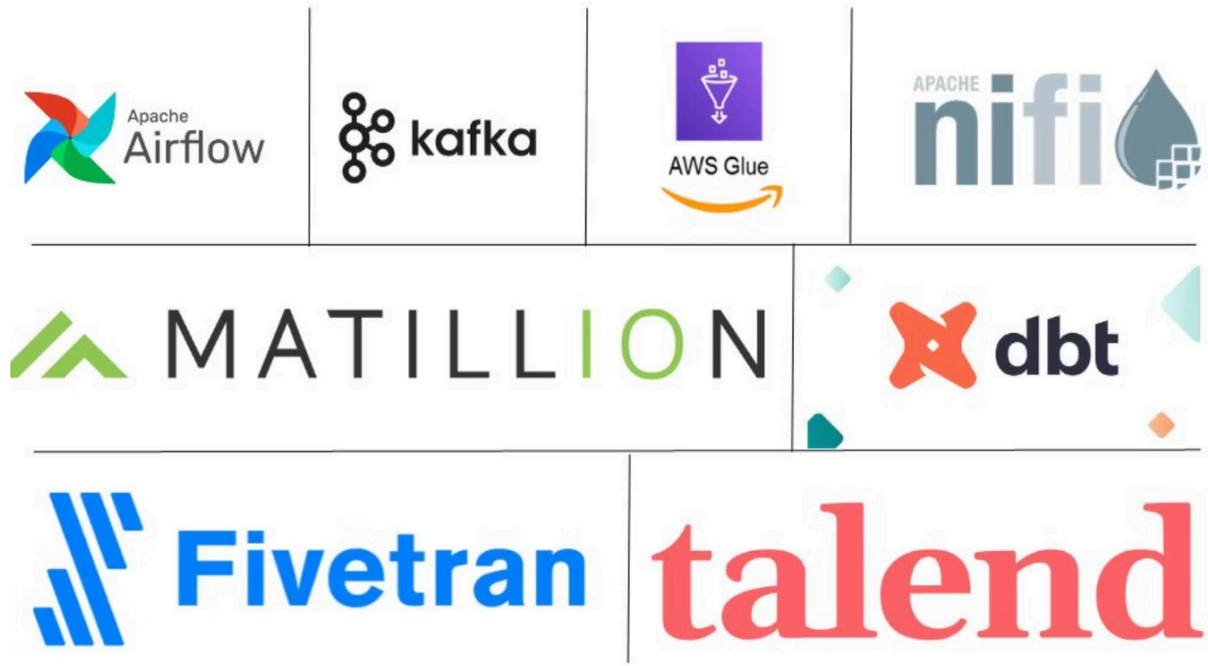
Es una plataforma de código abierto para la creación, programación y monitoreo de flujos de trabajo. Originalmente desarrollado por Airbnb, Airflow se ha convertido en una de las herramientas más populares para la automatización de flujos de datos, especialmente para la orquestación de trabajos por lotes y la gestión de pipelines de datos.

Apache NiFi

Es una plataforma de integración de datos y automatización de flujos de datos que permite la creación, gestión y monitoreo de flujos de datos de manera visual. Originalmente desarrollado por la NSA y luego donado a la Apache Software Foundation, NiFi se especializa en la automatización de flujos de datos en tiempo real, especialmente en entornos distribuidos.

Talend Data Fabrics

Es una plataforma de integración de datos que proporciona herramientas para la automatización de flujos de datos, la integración de sistemas, y la gestión de datos en la nube y on-premise. Talend ofrece soluciones tanto para el procesamiento por lotes como en tiempo real y es conocida por su capacidad de integración con diversas fuentes y destinos de datos.



Ventajas y desventajas de cada alternativa

Plataforma	Ventajas	Desventajas
Apache Airflow	- Orquestación flexible mediante DAGs (grafos acíclicos dirigidos) para definir tareas y dependencias. - Escalabilidad en clústeres distribuidos para manejar flujos de trabajo grandes. - Interfaz web intuitiva para monitorear y gestionar tareas, con capacidad de visualización de flujos de trabajo. - Amplia comunidad y ecosistema de plugins que facilita la integración con diversas herramientas y servicios.	- Enfoque principalmente para procesos por lotes. No optimizado para procesamiento en tiempo real. - Requiere configuración avanzada para escalar en entornos complejos. - Latencia alta en flujos con múltiples dependencias o tareas que requieren recursos elevados. - Curva de aprendizaje pronunciada, especialmente para usuarios nuevos.
Apache NiFi	- Procesamiento de datos en tiempo real con soporte para mover y transformar datos con baja latencia. - Interfaz visual intuitiva para diseñar y gestionar flujos de datos sin necesidad de escribir código. - Escalabilidad y resiliencia: capacidad de escalar horizontalmente y distribuir datos entre múltiples nodos. - Control del flujo de datos, priorización, enrutamiento condicional y gestión de errores.	- No ideal para flujos de trabajo extremadamente complejos con muchas dependencias. - Consumo de recursos relativamente alto (CPU, memoria), especialmente con grandes volúmenes de datos o flujos complejos. - Menos adecuado para orquestar tareas complejas o integraciones que requieren dependencias entre procesos. - Falta de características de control de versiones en los flujos, lo que puede dificultar la gestión de cambios y auditorías.
Talend Data Fabrics	- Solución integral para ETL, calidad de datos, gobernanza y integración de datos. - Soporte tanto para procesamiento en tiempo real como por lotes. - Facilidad de uso con una interfaz gráfica para diseñar flujos ETL, permitiendo la creación de pipelines de datos sin necesidad de escribir código extensivo. - Integración nativa con plataformas en la nube como AWS, Azure y Google Cloud, facilitando el trabajo con arquitecturas híbridas y sistemas en la nube.	- Costo elevado, especialmente en versiones comerciales. La licencia de Talend puede ser costosa para pequeñas empresas. - Menos enfoque en procesamiento en tiempo real, con un mayor enfoque en integración y calidad de datos. - Complejidad al configurar entornos de producción más grandes, especialmente cuando se necesitan características avanzadas. - Curva de aprendizaje moderada en comparación con plataformas como NiFi, ya que incluye más funcionalidades y herramientas dentro de una única plataforma.

Cuándo usar cada plataforma

Apache Airflow

1. Orquestación de flujos de trabajo por lotes: Ideal para automatizar tareas programadas y procesos que requieren orquestación de tareas con dependencias.
2. Automatización de pipelines ETL: Perfecto para proyectos que requieren mover, transformar y cargar grandes volúmenes de datos en intervalos regulares.
3. Flujos de trabajo complejos con múltiples dependencias: Ideal cuando las tareas deben ejecutarse en un orden específico o con dependencias complejas.
4. Integración con herramientas distribuidas: Cuando se necesita integrar con sistemas como Apache Spark o Hadoop para procesamiento distribuido.

Apache NiFi

1. Procesamiento en tiempo real: Perfecto para flujos de datos en tiempo real que requieren procesamiento y transferencia continua con baja latencia.
2. Integración de datos desde múltiples fuentes y destinos: Ideal cuando se necesita mover datos entre sistemas dispares (bases de datos, servicios web, APIs).
3. Control detallado del flujo de datos: Para gestionar tareas complejas como enrutamiento, filtrado, priorización y control de velocidad de datos.
4. Recuperación de fallos: Cuando se requiere una gestión resiliente de datos, con mecanismos avanzados de monitoreo y recuperación automática de fallos.

Talend Data Fabrics

1. Automatización de procesos ETL completos: Ideal para proyectos donde se requiere integración, transformación y carga de grandes volúmenes de datos.
2. Gobernanza y calidad de datos: Perfecto cuando se necesita asegurar que los datos sean limpios, consistentes y validados antes de su almacenamiento.
3. Integración de datos en la nube y on-premise: Ideal para mover datos entre entornos locales y la nube (AWS, Azure, Google Cloud).
4. Plataforma integral de datos: Para soluciones completas de gobernanza, calidad, y transformación de datos dentro de una plataforma centralizada.

Casos de uso y ejemplos

Plataforma	Caso de Uso	Ejemplo
Apache Airflow	Orquestación de flujos de trabajo por lotes	Automatización de un proceso ETL que extrae datos de una base de datos, los transforma y los carga en un Data Warehouse cada noche.
	Automatización de pipelines ETL	Realizar un proceso de análisis de datos donde se extraen logs de aplicaciones, se limpian y luego se almacenan en un sistema de análisis.
	Gestión de flujos de trabajo con dependencias complejas	Automatizar un flujo que descargue datos de una API, luego los procese en un sistema de análisis, y finalmente los almacene en un sistema de almacenamiento.
	Orquestación de procesos distribuidos	Orquestar tareas entre Apache Hadoop y Apache Spark para realizar procesamiento distribuido y almacenamiento de resultados.
Apache NiFi	Procesamiento de datos en tiempo real	Procesamiento en tiempo real de datos de sensores IoT en una fábrica para monitorear condiciones y enviar alertas en caso de anomalías.
	Integración de datos desde múltiples fuentes y destinos	Integración de datos de una base de datos, archivos CSV y una API externa, y luego envío de esos datos a un sistema de análisis en la nube.
	Enrutamiento y filtrado de datos	Procesar y filtrar logs de aplicaciones y redirigirlos según su tipo o severidad a diferentes destinos (bases de datos, sistemas de monitoreo).
	Control de flujo y priorización	Gestionar grandes volúmenes de datos de clientes y priorizar los más críticos para ser procesados antes que los menos urgentes.
Talend Data Fabrics	Automatización de procesos ETL completos	Migración de datos de múltiples bases de datos a un almacén de datos en la nube, incluyendo limpieza y transformación de datos.
	Gobernanza y calidad de los datos	Implementar un flujo de datos que valide y normalice datos antes de cargarlos en un sistema de análisis de datos en la nube, asegurando su calidad.
	Integración de datos en la nube y on-premise	Integrar datos de sistemas locales con plataformas en la nube (como AWS o Azure) para unificar información proveniente de diferentes fuentes.
	Manejo de grandes volúmenes de datos	Automatizar el procesamiento de grandes volúmenes de datos de clientes y productos, transformarlos y cargarlos en un sistema de Business Intelligence (BI).

Factores para considerar en la selección de una plataforma

La selección de una plataforma de automatización de flujos de datos depende de varios factores clave que deben ser evaluados para asegurar que la plataforma elegida se alinee con los objetivos del proyecto y las necesidades de la organización. A continuación, se describen los factores más importantes a considerar:

1. **Tipo de Flujo de Datos (Tiempo Real vs. Por Lotes)**
2. **Complejidad de los Flujos de Trabajo**
3. **Escalabilidad y Rendimiento**
4. **Facilidad de Uso y Configuración**
5. **Integración con Otras Herramientas y Sistemas**
6. **Soporte para Procesos de Transformación y Calidad de Datos**
7. **Costos de Implementación y Licenciamiento**

Arquitectura de Apache NiFi

La arquitectura de Apache NiFi está diseñada para ser flexible, escalable y distribuida, lo que permite gestionar flujos de datos de manera eficiente entre sistemas dispares. A continuación, se describen los principales componentes de su arquitectura y cómo interactúan entre sí.

Nodos de NiFi (NiFi Nodes)

NiFi opera en una arquitectura distribuida en la que el nodo principal (o nodo maestro) coordina las actividades, y los nodos adicionales (o nodos de trabajo) procesan y transfieren los datos.

Procesadores (Processors)

Son los bloques de construcción fundamentales de NiFi. Son responsables de mover, transformar, almacenar o enrutar los datos de una forma determinada. Cada procesador tiene una función específica que puede ser configurada por el usuario.

Conexiones (Connections)

Las conexiones en NiFi permiten el paso de datos de un procesador a otro. Son el vínculo entre los diferentes procesadores y permiten definir cómo los datos se mueven entre los componentes del flujo de datos.

Flujo de datos

Un flujo de datos en NiFi es una serie de procesadores conectados entre sí por conexiones que se encargan de mover, transformar y enrutar los datos de una fuente a un destino. Los flujos de datos en NiFi se diseñan de manera visual, lo que permite construir y gestionar de forma intuitiva.

Controladores

Son responsables de la gestión y administración de ciertos servicios en NiFi. Estos servicios pueden incluir la gestión de credenciales, el almacenamiento de información o la gestión de conexiones externas.

Flujos de Datos por Lotes

Los **flujos por lotes** están diseñados para procesar grandes volúmenes de datos en intervalos específicos. Son útiles en aplicaciones de **análisis de grandes conjuntos de datos**, **procesamiento de archivos históricos**, y tareas que requieren que los datos sean agrupados antes de ser procesados.

Pasos para Crear un Flujo de Datos por Lotes en NiFi:

- Iniciar NiFi**.**
- Configurar el Procesador de Entrada (por ejemplo, GetFile o ListFile)**.**
- Configurar un Procesador de Transformación (por ejemplo, ExtractText o ConvertRecord)**.**
- Almacenar los Datos Procesados.
- Programación de Procesos**.**
- Ejemplo de Flujo de Datos por Lotes**.**

Comparación entre Flujos en Tiempo Real y por Lotes en NiFi

Características	Flujos de Datos en Tiempo Real	Flujos de Datos por Lotes
Tipo de Datos	Datos que llegan continuamente (eventos, logs, sensores)	Datos que se acumulan en lotes (archivos, grandes conjuntos)
Procesamiento	Inmediato (en cuanto los datos llegan)	Programado (procesado a intervalos específicos)
Frecuencia	Datos se procesan de manera continua en tiempo real	Se procesan a intervalos predefinidos (diario, semanal, etc.)
Escalabilidad	Escalable para manejar grandes volúmenes de datos en tiempo real	Escalable para grandes volúmenes de datos en lotes
Uso Común	Monitoreo de sistemas en tiempo real, IoT, eventos en vivo	Procesamiento de archivos históricos, análisis de grandes volúmenes de datos
Ejemplo de Procesador	ListenHTTP, ListenKafka, GetFile	ListFile, GetFile, PutDatabaseRecord

Uso de procesadores para la ingesta, procesamiento y enrutamiento de datos

Apache NiFi ofrece una gran variedad de **procesadores** para gestionar flujos de datos. Estos procesadores permiten realizar tareas como **ingesta**, **procesamiento** y **enrutamiento** de datos. A continuación, se describen los diferentes tipos de procesadores disponibles para cada una de estas etapas, cómo se configuran y cómo se usan en un flujo de trabajo.

Procesadores Comunes para la Ingesta:

1 GetFile

Este procesador lee archivos de un directorio local y los convierte en FlowFiles para su posterior procesamiento.

2 ListFile

Este procesador lista archivos en un directorio sin leer el contenido. Genera FlowFiles con el nombre del archivo.

3 ListenHTTP

Permite que NiFi escuche solicitudes HTTP entrantes en un puerto determinado y convierte los datos recibidos en FlowFiles.

4 ConsumeKafka

Este procesador permite recibir mensajes de un tema Kafka. Cada mensaje en el tema se convierte en un FlowFile.

5 InvokeHTTP

Este procesador realiza solicitudes HTTP a una API externa o servicio web y convierte las respuestas en FlowFiles.

Controladores de Servicios en NiFi

Los controladores de servicios gestionan la configuración de servicios comunes que son utilizados por múltiples procesadores. Estos servicios incluyen conexiones a bases de datos, sistemas de archivos, servicios en la nube, entre otros. Configurar estos servicios correctamente es clave para la eficiencia de NiFi. Un controlador de servicio es un servicio compartido que se configura una sola vez y luego se puede reutilizar por varios procesadores en NiFi

Resumen de Configuración

Componente	Descripción	Configuración Típica
Controladores de Procesos	Agrupan y gestionan procesadores dentro de grupos de procesos para organizar flujos de trabajo complejos.	Crear un "Process Group" y configurar backpressure, manejo de errores y ejecución automática.
Controladores de Servicios	Gestionan conexiones y servicios compartidos, como bases de datos, almacenamiento en la nube, etc.	Configurar un servicio de base de datos (JDBC), HDFS, o conexión a S3, y habilitarlo para su uso en procesadores.
Políticas de Seguridad	Controlan la autenticación, autorización y acceso a recursos dentro de NiFi.	Configurar autenticación (LDAP, Kerberos), definir roles y permisos de acceso a procesadores y recursos.

Monitoreo del rendimiento en Apache NiFi

El monitoreo del rendimiento en Apache NiFi es crucial para garantizar que el sistema funcione de manera eficiente, especialmente cuando se manejan grandes volúmenes de datos o se ejecutan flujos complejos. NiFi ofrece diversas herramientas y funcionalidades para monitorear el rendimiento de los flujos de datos, los procesadores, los recursos del sistema y las métricas de tráfico en tiempo real.

- **Monitoreo de Flujos de Datos y Procesadores**
- **Monitoreo de Recursos del Sistema**
- **Control de Flujo y Optimización del Rendimiento**
- **Alertas y Notificaciones**
- **Integración con Herramientas de Análisis de Datos**

Elemento	Descripción	Herramientas
Monitoreo en la Interfaz Web	Visualización de flujos de datos, métricas de procesadores y estado de conexiones.	Interfaz web (Canvas) de NiFi
Métricas del Procesador	Métricas como la cantidad de FlowFiles procesados, bytes procesados, tiempo de ejecución, tasa de ejecución.	Panel de monitoreo y visualización de estado en la interfaz web
Recursos del Sistema	Monitoreo de uso de CPU, memoria y disco en la JVM que ejecuta NiFi.	JMX, Prometheus, Grafana
Backpressure	Control de flujo para evitar la sobrecarga de procesadores y conexiones.	Configuración en las conexiones entre procesadores
Alertas	Configuración de alertas para situaciones de error o rendimiento bajo.	Correo electrónico, integración con plataformas de monitoreo
Logs	Archivos de log con información detallada sobre la ejecución y errores de NiFi.	Archivos de log de NiFi (nifi-app.log, nifi-bootstrap.log)