Taller Práctico – Apache NiFi

Actividad utilizando Docker Desktop + Apache NiFi 1.28.1 + Python (ETL)

La siguiente actividad es un taller práctico para aprender a administrar datos por lotes (batch) utilizando las tecnologías ya mencionadas.

El objetivo es conocer la tecnología **Apache NiFi** como orquestador en la ingesta por lotes de archivos CSV, realizando algún proceso a estos datos y colocándolos en otra carpeta para finalmente ser procesados por Python con ETL.

Instalación: Apache NiFi 1.28.1 + Docker Desktop

Los requisitos previos son los siguientes:

- Docker Desktop instalado y en funcionamiento
- Acceso a terminal/línea de comandos
- Navegador web
- Carpetas de datos creadas en disco

Instalación – Paso 1: Verificar Docker Desktop

Debes abrir Docker Desktop (debe estar corriendo), abrir terminal o línea de comandos y ejecutar estos comandos de verificación:

```
docker –version
docker ps
```

Instalación – Paso 2: Descargar Apache NiFi

Ahora descargaremos la imagen de **Apache NiFi 1.28.1**. En la terminal, debes ejecutar:

```
docker pull apache/nifi:1.28.1
```

Este paso tomará bastante tiempo, entre 5 – 10 minutos, ya que la imagen pesa alrededor de 2 GB.

Instalación – Paso 3: Crear y Ejecutar contenedor Apache NiFi

A continuación crearemos el contenedor de Apache NiFi y lo ejecutaremos inmediatamente con algunas configuraciones especiales. Ejecuta este comando completo:

```
docker run --name apache-nifi \
  -p 8443:8443 -d \
  -e SINGLE_USER_CREDENTIALS_USERNAME=admin \
  -e SINGLE_USER_CREDENTIALS_PASSWORD=adminpassword12345 \
  -v "D:\Trabajo\Apache NiFi\datos\datos_crudos:/datos/datos_crudos:rw" \
  -v "D:\Trabajo\Apache NiFi\datos\datos_ingeridos:/datos/datos_ingeridos:rw" \
  apache/nifi:1.28.1
```

Esto generará un contenedor con las siguientes características:

- *--name apache-nifi*
  - Nombre del contenedor
- *-p 8443:8443*
  - Mapea puerto para acceso web
- *-d*
  - Ejecuta en segundo plano
- *-e SINGLE_USER_CREDENTIALS*
  - Configura credenciales personalizadas
- *-v "D:\...datos_crudos:/datos/datos_crudos:rw"*
  - Monta carpeta de datos crudos (lectura y escritura)
- *apache/nifi:1.28.1*
  - Imagen específica a usar

Explicación Detallada de la gestión de los Volúmenes. La sintaxis completa de Docker para montar volúmenes es:

```
-v "ruta_host:ruta_contenedor:opciones"
```

Se divide en 3 partes:

- Ruta del host (Windows): **D:\Trabajo\Apache NiFi\datos\datos_crudos** *(La ruta completa donde están los archivos en tu computadora)*
- Ruta dentro del contenedor: **/datos/datos_crudos** *(La ruta donde NiFi verá los archivos dentro del contenedor)*
- Opciones: **rw** *(lectura y escritura)*

Instalación – Paso 4: Inicializar Apache NiFi

Ahora tocan los siguientes pasos. Antes de abrir la interfaz de NiFi debes:

- **Esperar Inicialización -** Tiempo de espera: aproximadamente 5 minutos
- **Luego Abrir navegador web -** Ir a: https://localhost:8443/nifi
- **Advertencia de seguridad:** Normal con certificados autofirmados (Hacer clic en "Avanzado" → "Continuar a localhost")

Instalación – Paso 5: Iniciar Sesión en Apache NiFi

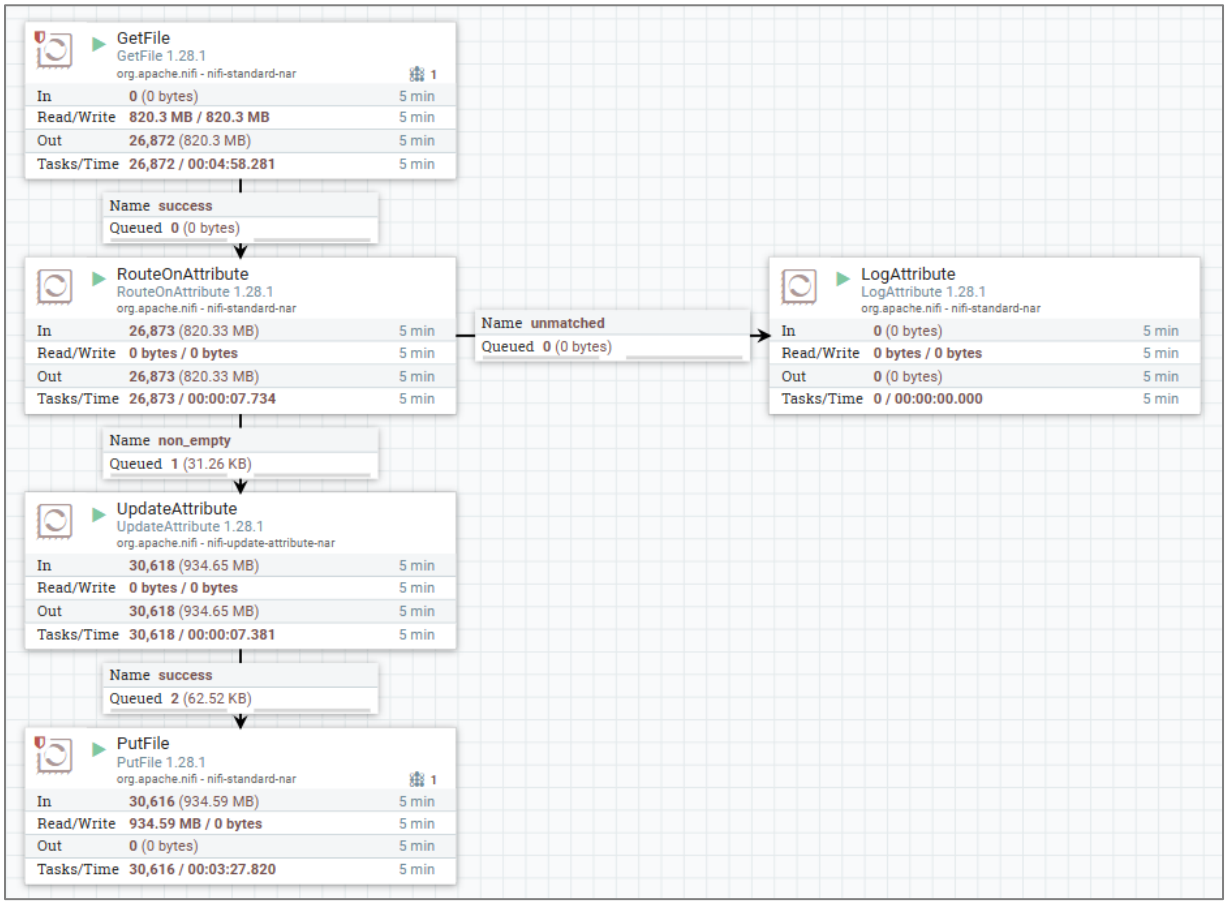
Una vez en el navegador web debes ingresar las siguientes credenciales:

- **Usuario -** admin
- **Contraseña –** adminpassword12345

Administración – Paso 1: Objetivos

Mover de forma automática el archivo CSV **`hospital_santiago_280.csv`** desde la carpeta de entrada **`datos_crudos`** a la carpeta de salida **`datos_ingeridos`**, validando que el archivo no esté vacío y renombrándolo de forma consistente.

Flujo ideal:



Instructor Felipe Cuevas

Administración – Paso 2: Requisitos y materiales

Los requisitos para empezar la actividad son:

- Apache NiFi 1.28.1 (se usará en Docker).
- Docker Desktop funcionando.
- Dos carpetas en tu computador (Windows/macOS/Linux):
    - …/datos/**datos_crudos** (entrada)
    - …/datos/**datos_ingeridos** (salida)
- Archivo de ejemplo: hospital_santiago_280.csv dentro de **datos_crudos**.

Administración – Paso 3: Crear un Process Group

Un Process Group es como una “carpeta” donde vive un mini-flujo completo. Ayuda a mantener todo ordenado. Para ordenar el diagrama haz lo siguiente:

- En el lienzo, clic derecho → Create process group.
- Nómbralo **01_ingesta_batch**.
- Doble clic para entrar.

Administración – Paso 4: Crear los Processors (procesadores)

Toma la imagen de referencia del flujo de procesadores que está en la página anterior y crea los siguientes procesadores:

1. **GetFile**
    a. Input Directory: /datos/datos_crudos
    b. File Filter: (?i).*\.csv (acepta .csv y .CSV)
    c. Keep Source File: true (ni mueve ni borra el original)
    d. Recurse Subdirectories: false (true si usas subcarpetas)
    e. Polling Interval: 0 sec
    f. Minimum File Age: 1 sec
    g. Ignore Hidden Files: true
2. **RouteOnAttribute**
    a. Routing Strategy: Route to Property name.
    b. Propiedad dinámica: Name (non_empty) – Value (${fileSize:gt(0)})
3. **LogAttribute**
    a. Configuración por defecto para capturar errores del **RouteOnAttribute**
4. **UpdateAttribute**
    a. mime.type = text/csv
    b. filename = ${filename:toLower():replaceAll('\\s','_')}
5. **PutFile**
    a. Directory: /datos/datos_ingeridos
    b. Conflict Resolution Strategy: replace
    c. Create Missing Directories: true

Proceso ETL – Paso 1: Python

Una vez que **Apache NiFi** haya ingerido, procesado y colocado los datos en la nueva carpeta, a través de Python realiza un pequeño proceso ETL para limpiar la información y entregar estadísticas sencillas sobre la misma:

1. **Elegir el archivo de trabajo**
    a. Identifica el CSV principal (por ejemplo, hospital_santiago_280.csv o el que tenga _ingested_... en el nombre).
    b. Confirma que abre sin errores y que ves una “tabla” con columnas y filas.
2. **Inspección rápida**
    a. Mira cuántas filas y columnas tiene.
    b. Revisa las primeras filas para entender cómo se llaman las columnas, qué tipo de valores hay (números, texto, fechas).
    c. Verifica si hay valores faltantes (nulos) en algunas columnas.
3. **Ajustar nombres de columnas**
    a. Convierte los nombres a un formato uniforme, por ejemplo tipo snake_case (minúsculas, sin espacios ni tildes).
    b. La idea es que todas las columnas queden “limpias” y fáciles de usar.
4. **Quitar duplicados**
    a. Elimina filas repetidas exactas para quedarte con un registro por fila.
    b. Anota cuántos duplicados se eliminaron (si los hay).
5. **Tratar valores faltantes (nulos)**
    a. Para columnas numéricas: decide si rellenas con algún valor razonable (por ejemplo, un promedio) o dejas el nulo si no corresponde rellenar.
    b. Para columnas de texto: decide si rellenas con una palabra como “desconocido” o prefieres dejar el nulo.
    c. Regla de oro: elige lo que tenga más sentido para tu contexto. Si no estás seguro, deja constancia en una nota dentro del notebook.
6. **Convertir tipos (si hace falta)**
    a. Fechas: convierte columnas que parecen fechas a un formato de fecha real.
    b. Números: si ves números guardados como texto (por comas o puntos), conviértelos a número.
7. **Genera Métricas y “foto” rápida del dataset.**
    a. Cuántas filas y columnas hay.
    b. Cuántos nulos por columna.
    c. Descripción básica de columnas numéricas (mínimo, máximo, promedio).
    d. Para columnas de texto: los 5 valores más frecuentes.
8. **Gráficos**
    a. Asegúrate de tener instaladas las librerías Matplotlib o Seaborn.
    b. Elige 1 o 2 columnas numéricas y 1 columna categórica.
9. **Tipos de gráfico recomendados**
    a. Histograma (numérica): para ver la distribución de una columna.
    b. Barras (categórica): para mostrar el Top 5 o Top 10 de categorías más frecuentes.
    c. Línea (serie de tiempo): para observar la evolución por fecha.
    d. Dispersión (dos numéricas): para ver relación entre dos variables.