

Parte 1: Introducción

Se realizará un flujo para introducir en la herramienta.

Acceder a la interfaz de NiFi en <https://localhost:8443/nifi>

Es necesario que primero se haya configurado el túnel con la redirección de puertos.

Para iniciar sesión en NiFi:

Usuario: admin

Contraseña: practiconifi

Definición de carpeta de salida.

Se debe crear un directorio con permisos donde almacenar los datos generados

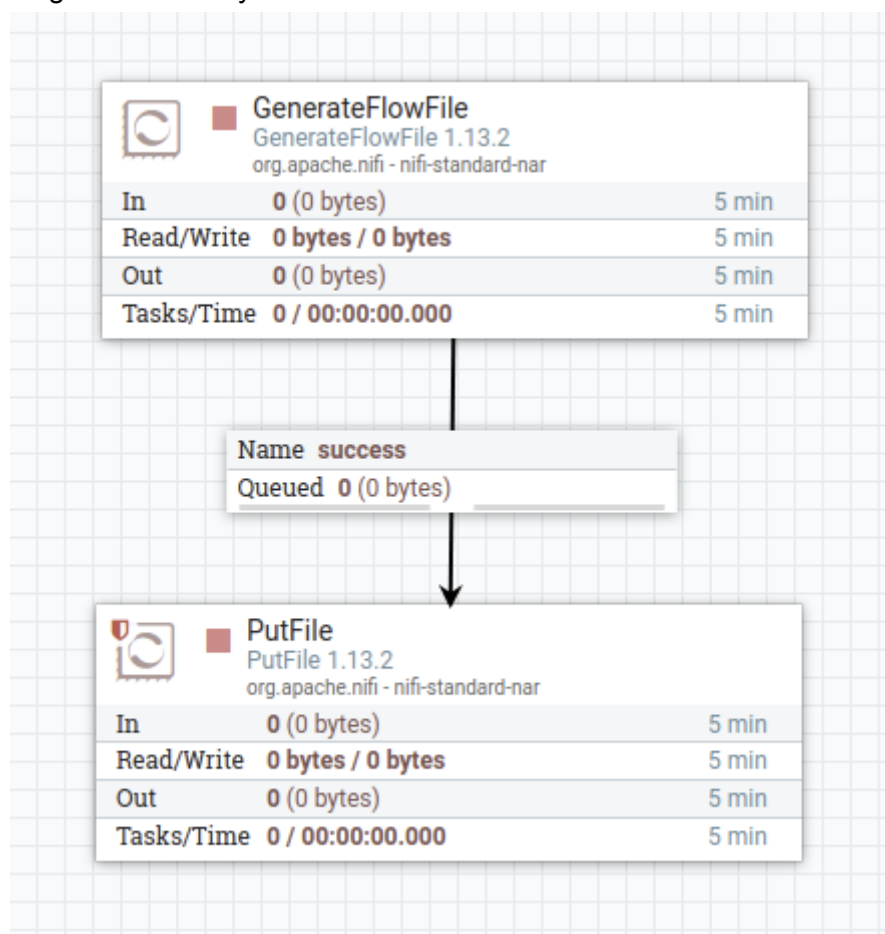
Estando en el home del usuario (/home/azureuser/) ejecutar:

```
mkdir -p practico_nifi/out
```

```
chmod 777 practico_nifi/out
```

Creación del flujo

El flujo deberá generar datos y almacenarlos en un directorio local.



1. Agregar un processor GenerateFlowFile y configurarlo de la siguiente manera:
 - En la pestaña Properties, configurar 'Custom Text' en: esto es una prueba \${now()}
2. Agregar un processor PutFile y configurarlo de la siguiente manera:
 - En la pestaña Properties, configurar 'Directory' en: /home/azureuser/practico_nifi/out
 - En la pestaña Settings, seleccionar para auto terminar en failure y success.
3. Generar una conexión entre el GenerateFlowFile y el PutFile
4. Iniciar solamente el GenerateFlowFile por unos segundos y **detenerlo**.
5. Examinar el queue de la conexión entre los dos processors:
 - Click derecho, list queue.
 - Seleccionar un flowfile y analizar las opciones, por ejemplo 'View'.
6. Iniciar únicamente el processor PutFile y analizar el directorio out.
7. Verificar el directorio de salida con la terminal.
8. **Detener todo el flujo.**
9. Borrar los archivos:

```
rm -f /home/azureuser/practico_nifi/out/*
```

Parte 2: Procesamiento de info de GPS

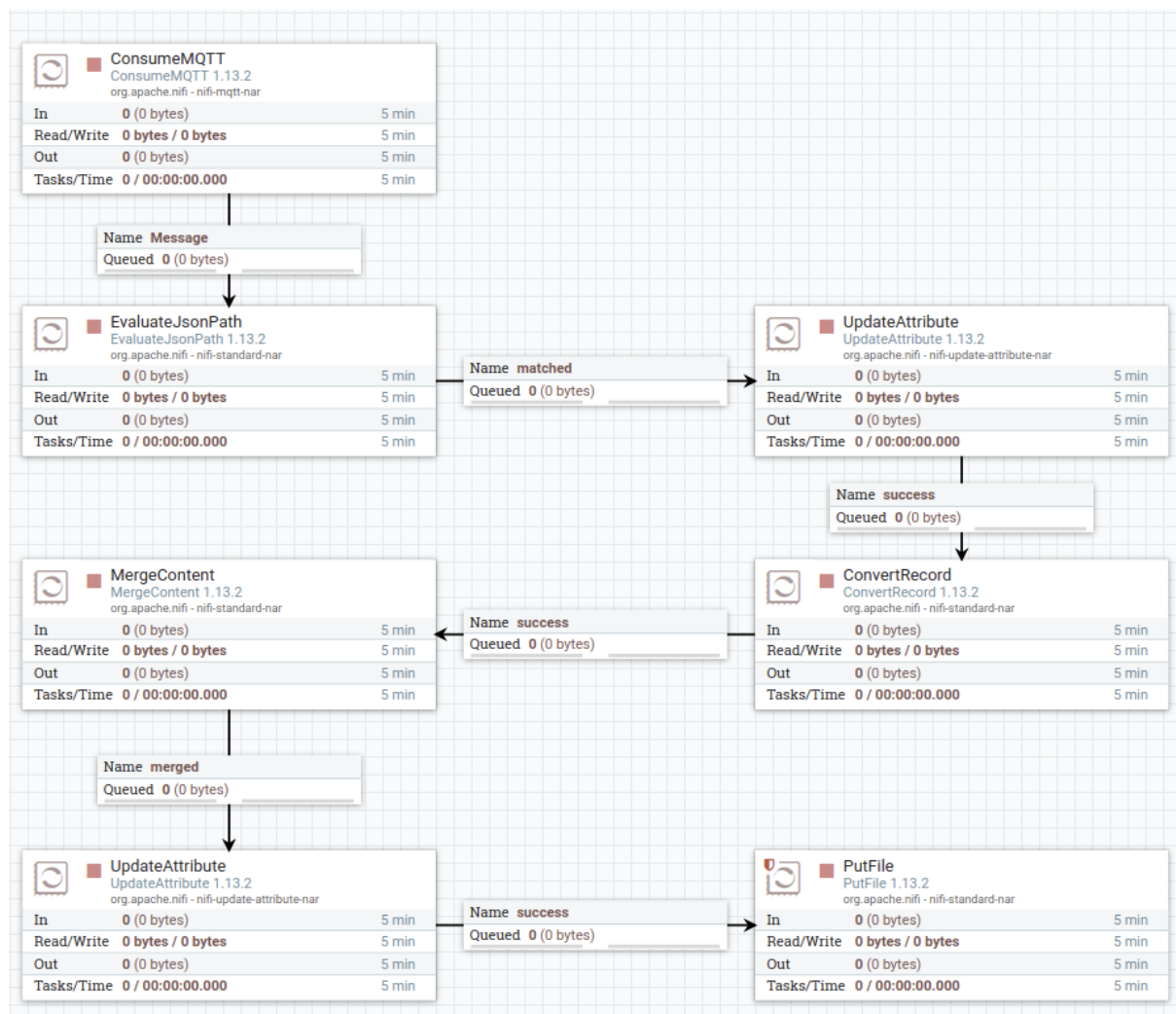
Definición de carpeta de salida.

Se debe crear un directorio con permisos donde almacenar los datos generados
Estando en el home del usuario (/home/azureuser/) ejecutar:

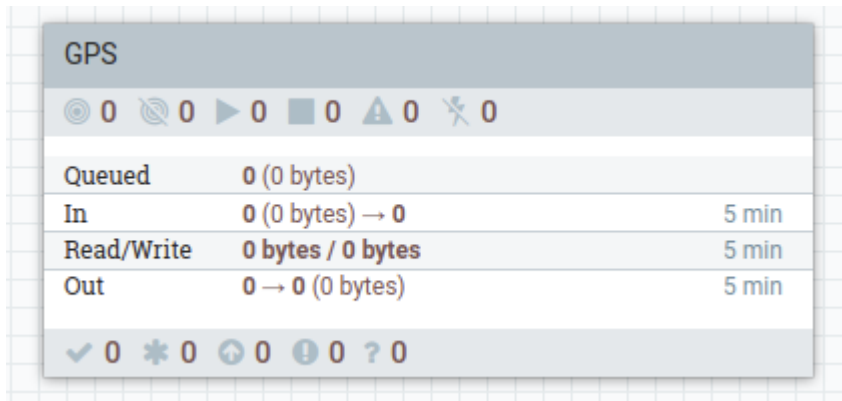
```
mkdir -p practico_nifi/gps/out  
chmod 777 practico_nifi/gps/out
```

Creación del flujo

El flujo a crear tomará datos JSON de MQTT, los transformará y agrupará en CSV para almacenar en directorios locales.



1. Crear un Process Group GPS y acceder al mismo



2. Agregar un processor ConsumeMQTT y configurarlo de la siguiente forma:
 - a. En Properties:
 - i. Broker URI: tcp://localhost:1883
 - ii. Topic Filter: gps
 - iii. Max Queue Size: 100
 - b. En Settings:
 - i. Auto terminar parse.failure
3. Agregar un processor EvaluateJsonPath para crear un atributo con el id_vehiculo y configurarlo de la siguiente forma:
 - a. Conectar desde ConsumeMQTT con la relación Message
 - b. En Properties:
 - i. Destination: flowfile-attribute
 - ii. Crear una nueva property id_vehiculo: \$.id_vehiculo
 - c. En Settings:
 - i. Auto terminar failure y unmatched
4. Agregar un processor UpdateAttribute para crear un atributo con el path del directorio final y configurarlo de la siguiente forma:
 - a. Conectar desde EvaluateJsonPath con la relación matched
 - b. En Properties:
 - i. Crear una nueva property path: /home/azureuser/practico_nifi/gps/out/\${id_vehiculo}
5. Para transformar el JSON en un CSV, agregar un processor ConvertRecord y configurarlo de la siguiente forma:
 - a. Conectar desde UpdateAttribute con la relación success
 - b. En Properties:
 - i. En Record Reader crear uno nuevo y elegir JsonTreeReader.
 - ii. En Record Writer crear uno nuevo y elegir CSVRecordSetWriter
 - c. En Settings:
 - i. Auto terminar failure.
6. Configurar los Controller Services. Dar click derecho en la pantalla y elegir Configure:
 - a. Seleccionar la opción de configurar el CSVRecordSetWriter y cambiar las propiedades:

- i. Include Header Line: false.
- ii. Schema Access Strategy: Use 'Schema Text' Property
- iii. Schema Text:

```
{
  "name": "MyClass",
  "type": "record",
  "namespace": "com.acme.avro",
  "fields": [
    {
      "name": "timestamp",
      "type": "float"
    },
    {
      "name": "latitud",
      "type": "float"
    },
    {
      "name": "longitud",
      "type": "float"
    },
    {
      "name": "velocidad",
      "type": "int"
    },
    {
      "name": "id_vehiculo",
      "type": "string"
    }
  ]
}
```

- b. Habilitar los dos servicios: CSVRecordSetWriter y JsonTreeReader

7. Para agrupar datos agregar un processor MergeContent y configurarlo de la siguiente forma:

- a. Conectar desde ConvertRecord con la relación success
- b. En Properties:
 - i. Correlation Attribute Name: id_vehiculo
 - ii. Minimum Number of Entries: 5
 - iii. Max Bin Age: 3 seconds
 - iv. Maximum number of Bins 2
- c. En Settings:
 - i. Auto terminar failure y original

8. Agregar un processor UpdateAttribute para cambiar el nombre del archivo a escribir.

- a. Conectar desde MergeContent con la relación merged
- b. En Properties:
 - i. Crear una propiedad filename: \${UUID()}.csv

9. Para almacenar resultados crear un processor PutFile y configurarlo de la siguiente forma:

- a. Conectar desde UpdateAttribute con la relación success
- b. En Properties:
 - i. Directory: \${path}
- c. En Settings:
 - i. Auto terminar success y failure

10. Iniciar el generador de datos de gps.
11. Iniciar el flujo de a 1 processor, analizando los datos a medida que avanzan.
12. Analizar el resultado en el directorio de salida.
13. Cancelar el simulador.
14. Detener todo el flujo.
15. Eliminar el los datos ingestados:
`rm -rf /home/azureuser/practico_nifi/gps/out/*`