

# Apache NiFi

Pedro González Fernández



30 de septiembre de 2025

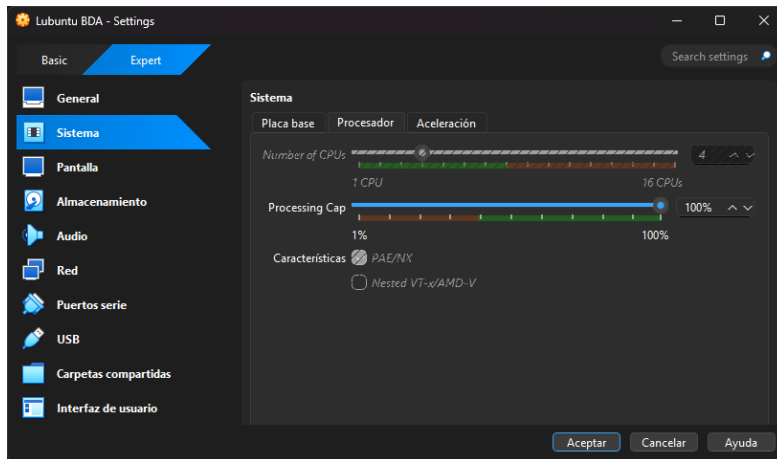
## Índice

<b>1. Lubuntu 24.04</b>	<b>2</b>
<b>2. Preparación del entorno</b>	<b>3</b>
<b>3. Manejo de ficheros</b>	<b>9</b>
<b>4. Uso de plantillas en NiFi</b>	<b>17</b>
<b>5. Practicando con grupos</b>	<b>19</b>

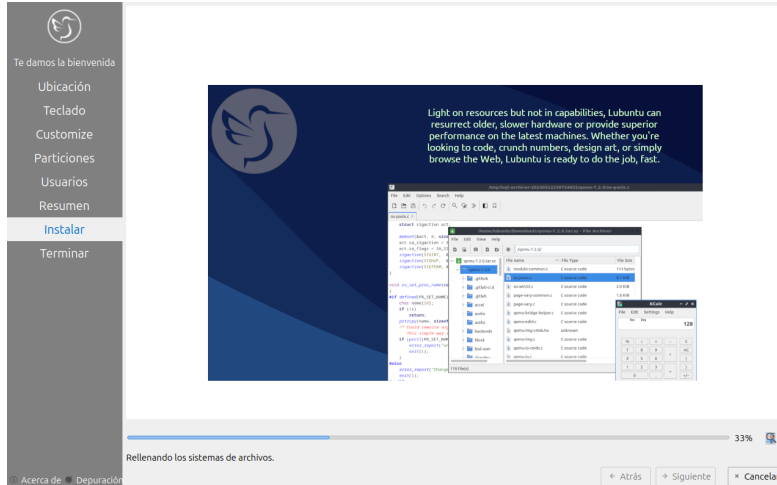
## 1. Ubuntu 24.04

En esta práctica instalaremos y configuraremos el sistema operativo **Lubuntu 24.04** en **VirtualBox**. Para ello debemos descargar la ISO desde la [página oficial](#).

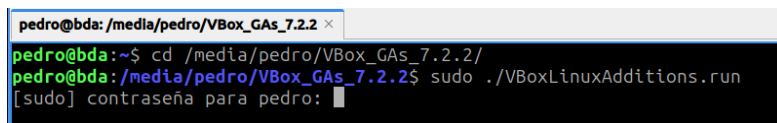
Una vez descargada, creamos la máquina virtual en VirtualBox. En este caso, se le asignaron 8 GB de RAM, 4 procesadores y 50 GB de disco reservado dinámicamente. Es recomendable activar la opción **EFI** y habilitar el **PAE/NX** en la configuración del sistema:



Tras esto, iniciamos la máquina e instalamos el sistema de manera habitual:



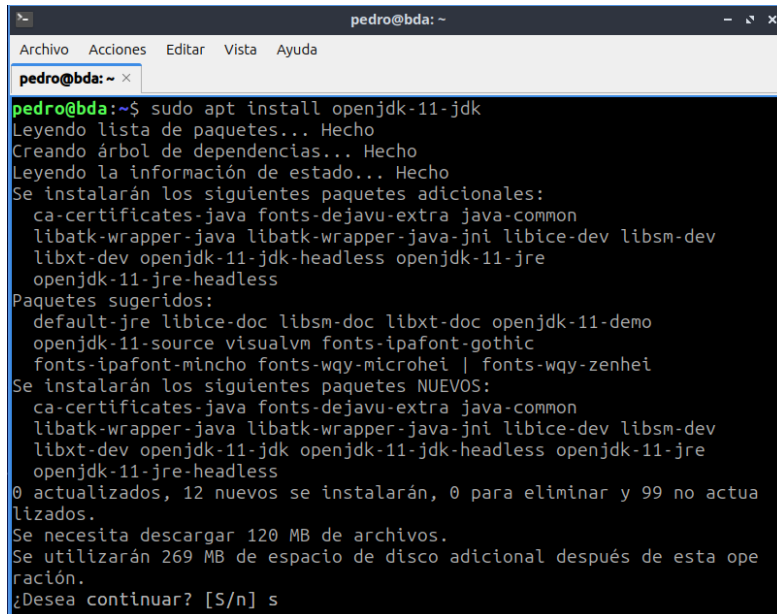
Finalmente, instalamos las **Guest Additions** y tendremos la máquina lista para trabajar:



## 2. Preparación del entorno

En esta sección prepararemos el entorno para instalar y configurar **Apache NiFi** en la máquina virtual con Ubuntu 24.04.

El primer paso consiste en instalar la **JDK**:




```

pedro@bda: ~$ sudo apt install openjdk-11-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ca-certificates-java fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni libice-dev libsm-dev
  libxt-dev openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless
Paquetes sugeridos:
  default-jre libice-doc libsm-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
  ca-certificates-java fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni libice-dev libsm-dev
  libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 99 no actua
lizados.
Se necesita descargar 120 MB de archivos.
Se utilizarán 269 MB de espacio de disco adicional después de esta ope
ración.
¿Desea continuar? [S/n] s

```

Comprobamos que la instalación se ha realizado correctamente verificando la versión instalada:



```

pedro@bda:~$ sudo update-alternatives --config java
Existe 1 opción para la alternativa java (que provee /usr/bin/java).

  Selección   Ruta                                          Prioridad
Estado
-----
* 0          /usr/lib/jvm/java-11-openjdk-amd64/bin/java  1111
modo automático
  1          /usr/lib/jvm/java-11-openjdk-amd64/bin/java  1111
modo manual

Pulse <Intro> para mantener el valor por omisión [*] o pulse un núme
ro de selección:
pedro@bda:~$ java -version
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment (build 11.0.28+6-post-Ubuntu-1ubuntu124.
.1)
OpenJDK 64-Bit Server VM (build 11.0.28+6-post-Ubuntu-1ubuntu124.04.
mixed mode, sharing)
pedro@bda:~$ javac -version
javac 11.0.28
pedro@bda:~$

```

A continuación, configuramos las variables de entorno de **Java**. Desde el directorio **home**, editamos el archivo **.bashrc** y añadimos las siguientes líneas al final:

```

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH

```

En paralelo, instalamos la versión **JDK 21** para que NiFi utilice esta versión, manteniendo la JDK 11 como predeterminada en el sistema:

```
pedro@bda:~$ sudo apt install openjdk-21-jdk
[sudo] contraseña para pedro:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  openjdk-21-jdk-headless openjdk-21-jre openjdk-21-jre-headless
Paquetes sugeridos:
  openjdk-21-demo openjdk-21-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
  openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre openjdk-21-jre-headless
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 99 no actualizados.
Se necesita descargar 131 MB de archivos.
```

Posteriormente comprobamos las rutas de instalación y configuramos **Java 11** como versión predeterminada:

```
pedro@bda:~$ ls -ld /usr/lib/jvm/java-21-openjdk-amd64/
/usr/lib/jvm/java-21-openjdk-amd64/
pedro@bda:~$ sudo update-alternatives --config java
Existen 2 opciones para la alternativa java (que provee /usr/bin/java).

Selección Ruta Prioridad Estado
-----
* 0 /usr/lib/jvm/java-21-openjdk-amd64/bin/java 2111 modo automático
1 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 modo manual
2 /usr/lib/jvm/java-21-openjdk-amd64/bin/java 2111 modo manual

Pulse <Intro> para mantener el valor por omisión [*] o pulse un número de selección: 1
update-alternatives: utilizando /usr/lib/jvm/java-11-openjdk-amd64/bin/java para proveer /usr/bin/java (java) en modo manual
pedro@bda:~$ sudo update-alternatives --config javac
Existen 2 opciones para la alternativa javac (que provee /usr/bin/javac).

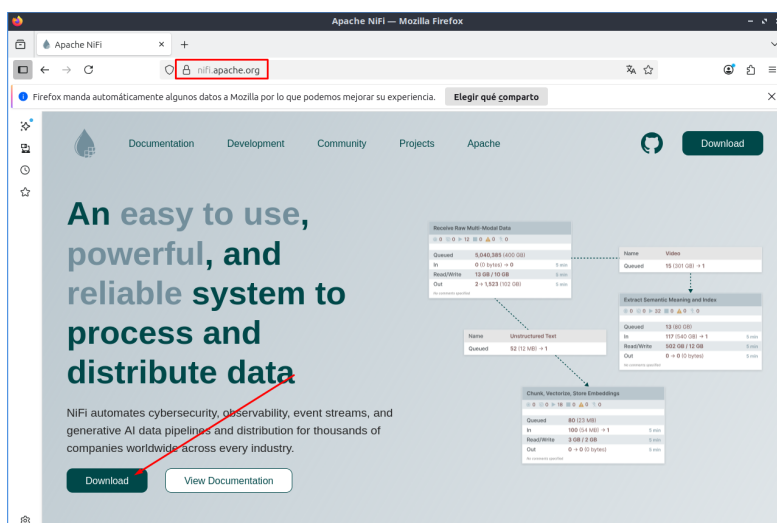
Selección Ruta Prioridad Estado
-----
* 0 /usr/lib/jvm/java-21-openjdk-amd64/bin/javac 2111 modo automático
1 /usr/lib/jvm/java-11-openjdk-amd64/bin/javac 1111 modo manual
2 /usr/lib/jvm/java-21-openjdk-amd64/bin/javac 2111 modo manual

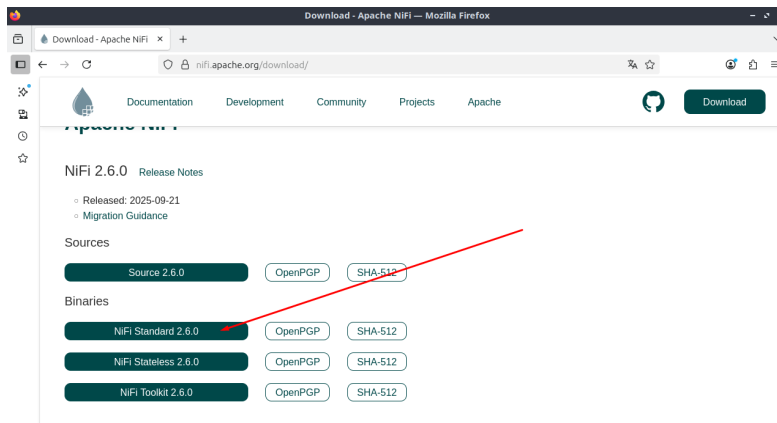
Pulse <Intro> para mantener el valor por omisión [*] o pulse un número de selección: 1
update-alternatives: utilizando /usr/lib/jvm/java-11-openjdk-amd64/bin/javac para proveer /usr/bin/javac (javac) en modo manual
```

*Nota: seleccionar el número correspondiente a la versión 11.*

```
pedro@bda:~$ java -version
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment (build 11.0.28+6-post-Ubuntu-1ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 11.0.28+6-post-Ubuntu-1ubuntu124.04.1, mixed mode, sharing)
```

A continuación descargamos **Apache NiFi** desde la [página oficial](https://nifi.apache.org):





Una vez descargado, lo descomprimos:

```
pedro@bda:~/Descargas$ ls
nifi-2.6.0-bin.zip
pedro@bda:~/Descargas$ unz
unzip  unzipsfx  unzstd
pedro@bda:~/Descargas$ unzip nifi-2.6.0-bin.zip
Archive:  nifi-2.6.0-bin.zip
  creating: nifi-2.6.0/
  creating: nifi-2.6.0/extensions/
  creating: nifi-2.6.0/lib/
  creating: nifi-2.6.0/lib/bootstrap/
  creating: nifi-2.6.0/lib/aspectj/
  creating: nifi-2.6.0/conf/
  creating: nifi-2.6.0/bin/
  creating: nifi-2.6.0/docs/
  creating: nifi-2.6.0/docs/html/
```

Y lo movemos a un directorio más adecuado:

```
pedro@bda:~/Descargas$ ls
nifi-2.6.0  nifi-2.6.0-bin.zip
pedro@bda:~/Descargas$ sudo mv nifi-2.6.0 /opt/nifi
[sudo] contraseña para pedro:
pedro@bda:~/Descargas$ ls /opt/nifi/
bin  conf  docs  extensions  lib  LICENSE  NOTICE  python  README
pedro@bda:~/Descargas$
```

Definimos también las variables de entorno de NiFi, editando nuevamente el archivo `.bashrc` y añadiendo lo siguiente:

```
export NIFI_HOME=/opt/nifi
export PATH=$NIFI_HOME/bin:$PATH
```

Ahora intentamos arrancar NiFi:

```
pedro@bda:~$ nifi.sh start
nifi.sh: JAVA_HOME is not valid: /usr/lib/jvm/java-11-openjdk-amd64
pedro@bda:~$
```

Observamos que aparece un error, ya que la versión 11 de Java no es compatible con esta versión de NiFi.

Para solucionarlo, editamos el archivo `nifi-env.sh` e indicamos que NiFi utilice la JDK 21:

```
JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
```

```
pedro@bda:~$ sudo nano /opt/nifi/bin/nifi-env.sh
pedro@bda:~$ nifi.sh start

JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
NIFI_HOME=/opt/nifi
```

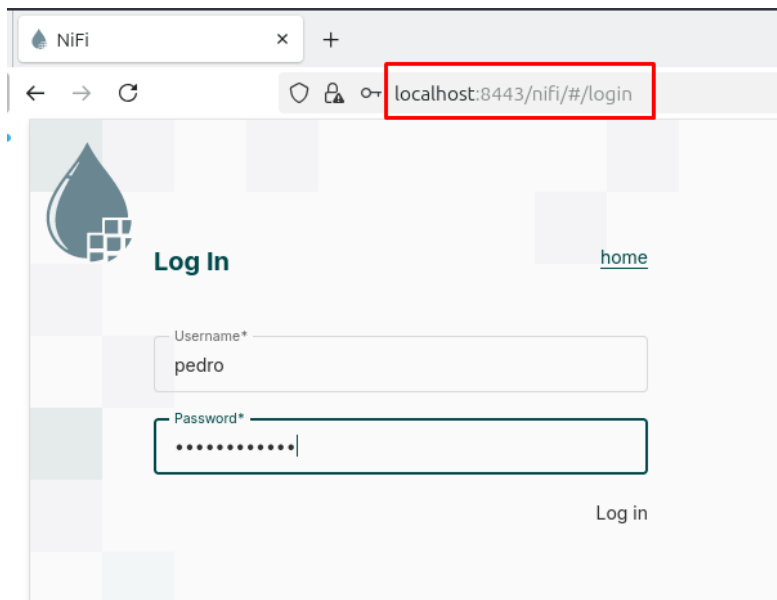
Como se aprecia, ahora NiFi arranca sin problemas. Lo detenemos con `nifi.sh stop` y configuramos un usuario y contraseña:

```
pedro@bda:~$ nifi.sh set-single-user-credentials pedro 112233445566

JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
NIFI_HOME=/opt/nifi

Login Identity Providers Processed [/opt/nifi/conf/login-identity-providers.xml]
```

Si lo iniciamos nuevamente y accedemos a **`https://localhost:8443`**, llegamos al panel de inicio de sesión de NiFi, donde podemos entrar con las credenciales configuradas:



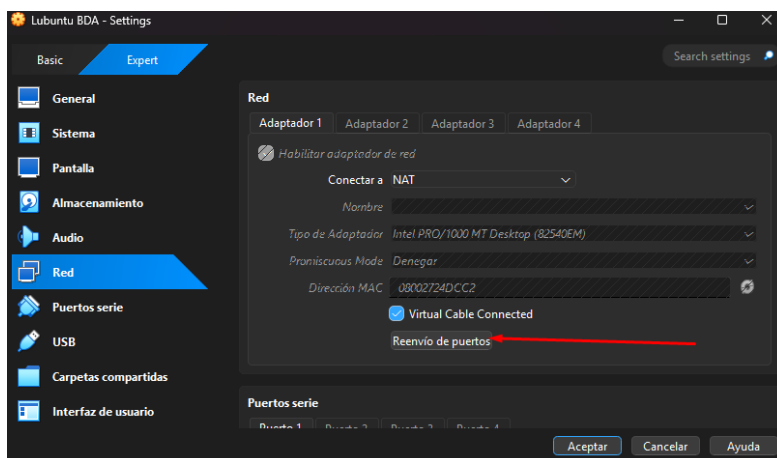
Para facilitar la gestión y el acceso desde el host, modificamos el archivo `nifi.properties`, ubicado en `/opt/nifi/conf`, de la siguiente forma:

```
nifi.web.http.host=0.0.0.0
nifi.web.http.port=8081
nifi.web.http.network.interface.default=
#####
#nifi.web.https.host=localhost
#nifi.web.https.port=8443
nifi.web.https.network.interface.default=
nifi.web.https.application.protocols=h2 http/1.1
```

```
nifi.security.autoreload.enabled=false
nifi.security.autoreload.interval=10 secs
#nifi.security.keystore=./conf/keystore.p12
nifi.security.keystore.certificate=
nifi.security.keystore.privateKey=
#nifi.security.keystoreType=PKCS12
#nifi.security.keystorePasswd=1413faac7ea20896a5090aeb0c565e89
#nifi.security.keyPasswd=1413faac7ea20896a5090aeb0c565e89
#nifi.security.truststore=./conf/truststore.p12
nifi.security.truststore.certificate=
#nifi.security.truststoreType=PKCS12
#nifi.security.truststorePasswd=dde7ce9573aeba3961b06ca4f5e8d217
```

```
nifi.remote.input.host=
nifi.remote.input.secure=false
nifi.remote.input.socket.port=
```

Dado que la máquina virtual está configurada en modo **NAT**, no es posible acceder directamente desde el exterior. Para solucionarlo, configuramos un reenvío de puertos en VirtualBox, redirigiendo el puerto 8081 de la máquina virtual al 8080 de la máquina anfitriona:



Nombre	Protocolo	IP anfitrión	Puerto anfitrión	IP invitado	Puerto invitado
NiFi	TCP		8080		8081

*Nota: el puerto anfitrión corresponde al host, mientras que el puerto invitado corresponde al servicio de NiFi en la máquina virtual.*





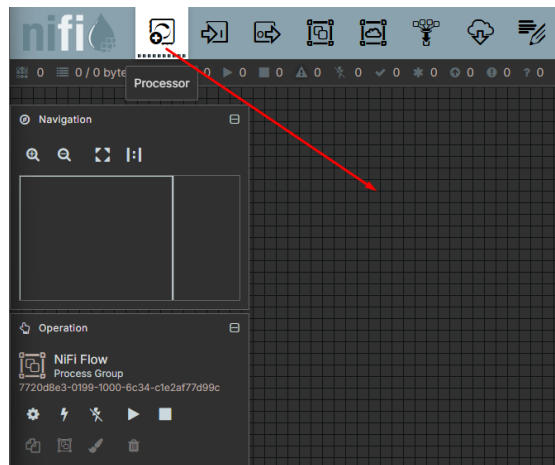
### 3. Manejo de ficheros

A continuación aprenderemos a manejar ficheros de forma sencilla con NiFi, creando un par de flujos para generar, renombrar y mover archivos.

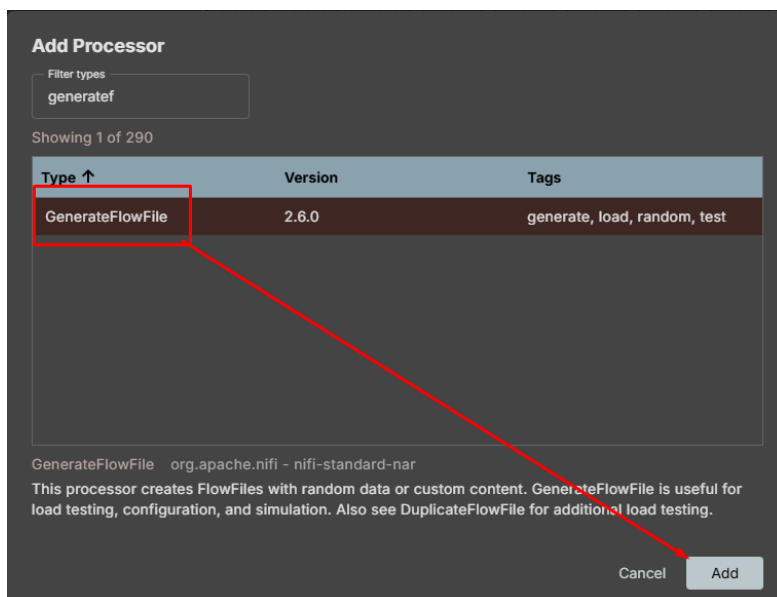
Primero creamos los directorios de trabajo en la máquina:

```
pedro@bda:~$ mkdir -p workspace/entradaNifi
pedro@bda:~$ mkdir -p workspace/salidaNifi
pedro@bda:~$ ll workspace/
total 16
drwxrwxr-x  4 pedro pedro 4096 sep 24 15:39 ./
drwxr-x--- 15 pedro pedro 4096 sep 24 15:39 ../
drwxrwxr-x  2 pedro pedro 4096 sep 24 15:39 entradaNifi/
drwxrwxr-x  2 pedro pedro 4096 sep 24 15:39 salidaNifi/
pedro@bda:~$
```

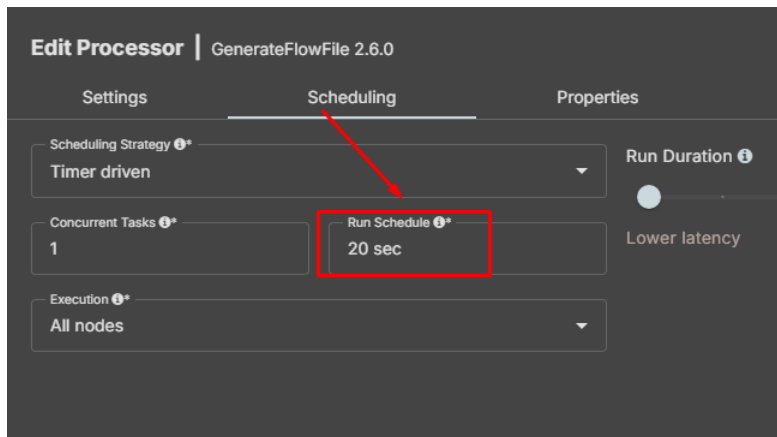
Accedemos a NiFi y comenzamos con el primer flujo. Arrastramos un proceso al área de trabajo:



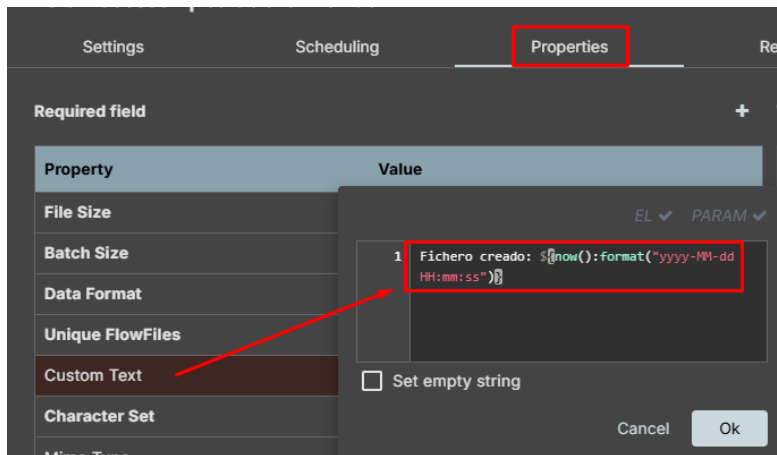
Seleccionamos el tipo de proceso. En este caso, **GenerateFlowFile**, que genera archivos:



Hacemos doble clic en el proceso y lo configuramos. En la pestaña *Scheduling*, establecemos que se ejecute cada 20 segundos:

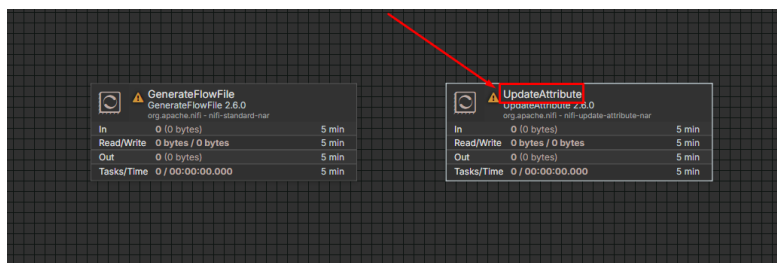


En la pestaña *Properties*, en el campo *Custom Text*, escribimos lo siguiente:

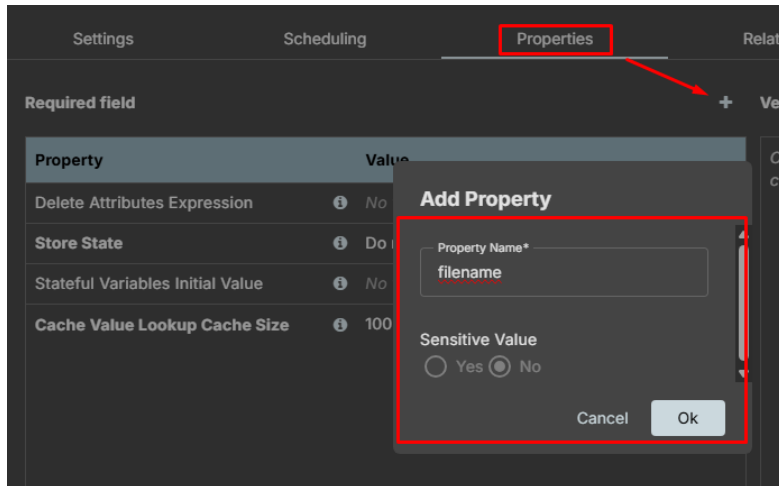


Esto hará que cada archivo generado contenga dicho texto junto con la fecha y hora de creación.

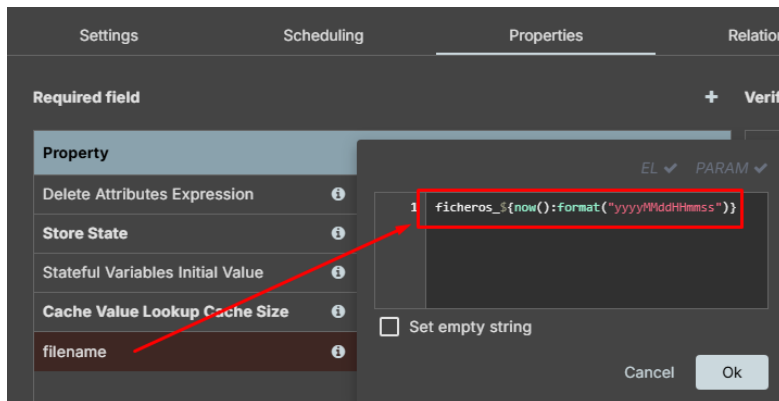
Luego añadimos otro proceso, **UpdateAttribute**, para modificar atributos del fichero creado:



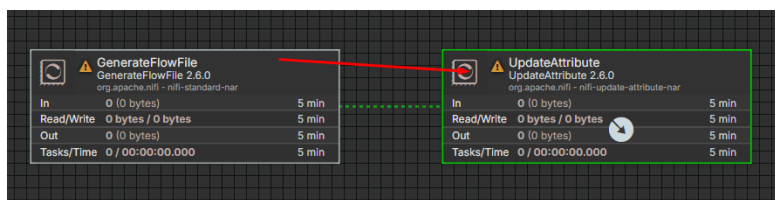
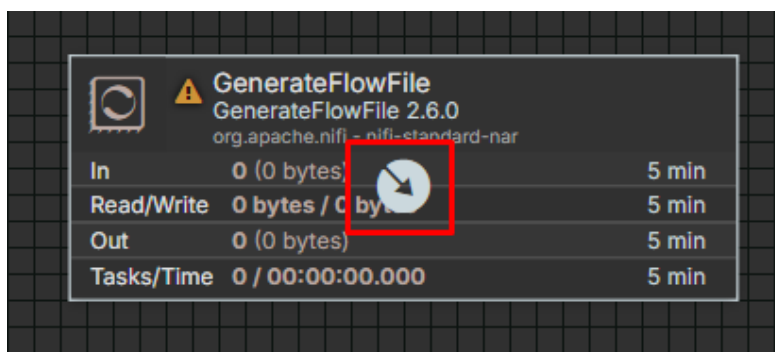
En sus propiedades, añadimos un nuevo atributo haciendo clic en el icono +:



Lo nombramos `filename` (atributo del nombre de archivo) y le damos el siguiente contenido:

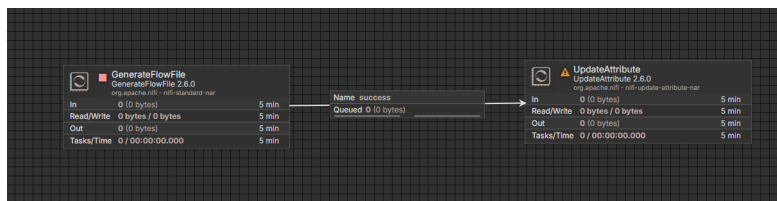


De esta manera, el nombre del archivo estará compuesto por la fecha y hora actuales. Conectamos ambos procesos arrastrando la flecha desde el primero hacia el segundo:

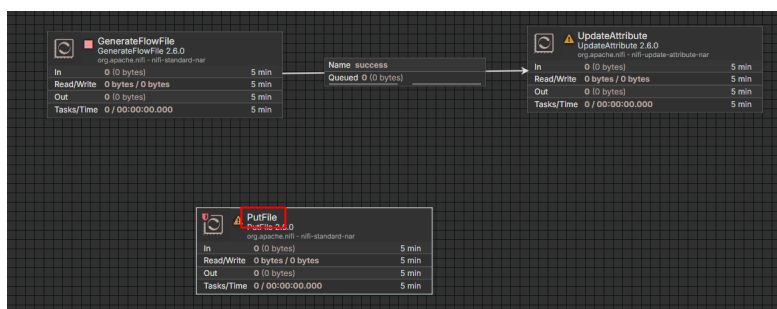


En la configuración de la conexión seleccionamos la prioridad **FIFO** (**F**irst **I**n **F**irst **O**ut):

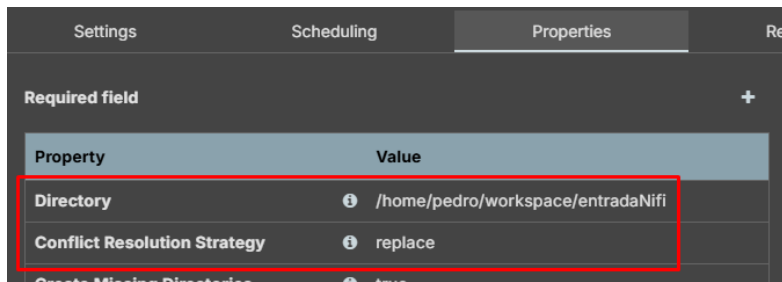
Así obtenemos un flujo como el siguiente:



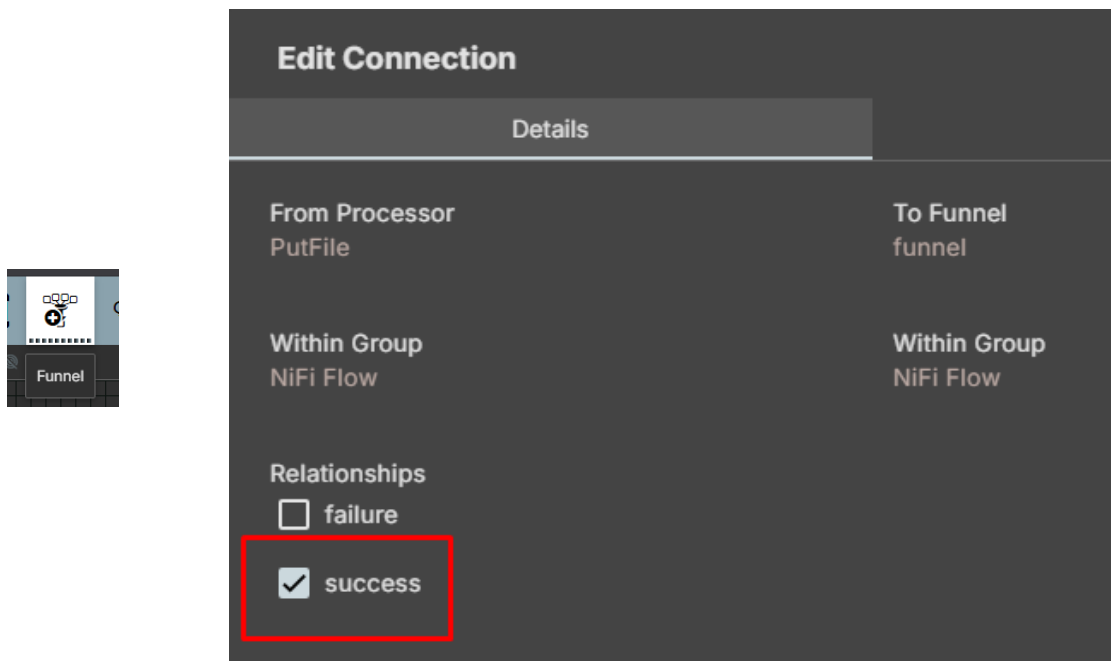
Después añadimos un proceso **PutFile**, que colocará el fichero en el directorio de entrada:



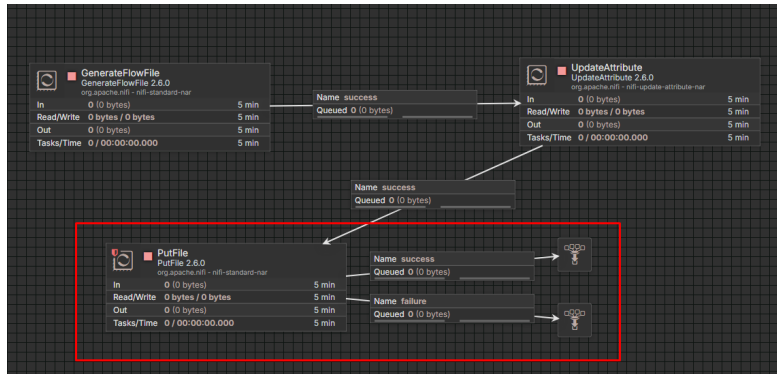
En sus propiedades configuramos los campos **Directory** y **Conflict Resolution Strategy**:



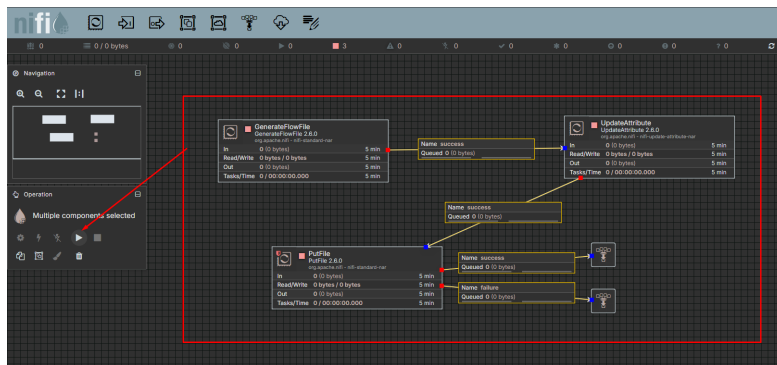
Para validar los resultados, añadimos dos **Funnel**: uno para los ficheros exitosos (marcando *Success*) y otro para los fallidos (marcando *Failure*):



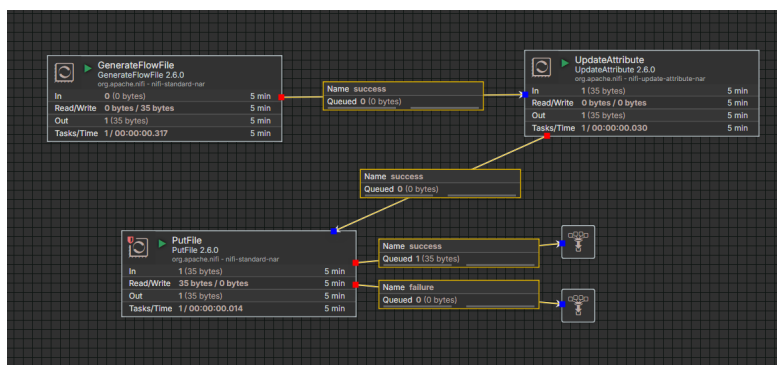
El resultado debería ser similar a este:



Ya tenemos el primer flujo listo. Lo seleccionamos completo (con **Shift**) y lo arrancamos:



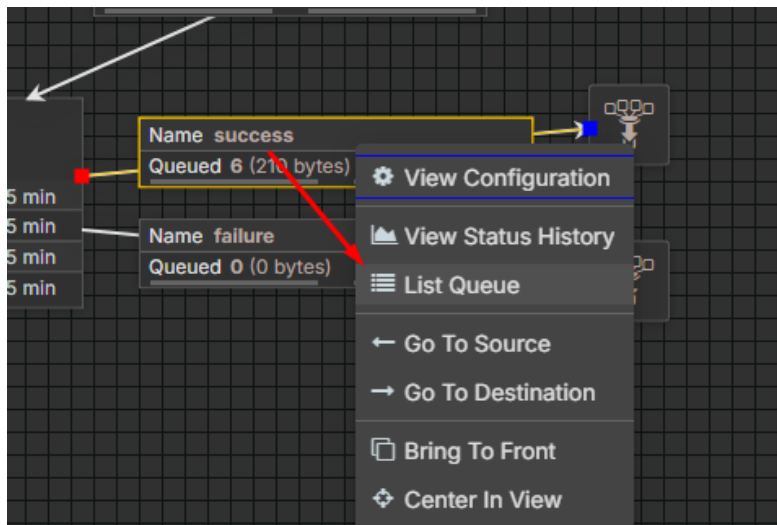
Los procesos comenzarán a ejecutarse automáticamente:



En el directorio configurado de entrada se irán generando ficheros:

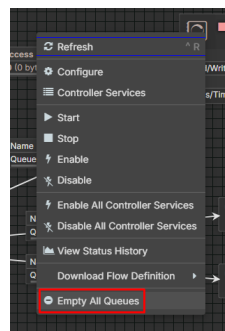
```
pedro@bda:~/workspace$ ll entradaNifi/
total 24
drwxrwxr-x 2 pedro pedro 4096 sep 24 16:14 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:13 ficheros_20250924161352
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:14 ficheros_20250924161412
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:14 ficheros_20250924161432
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:14 ficheros_20250924161452
```

Podemos comprobar también la cola de *Success*:



Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Permalized
1	ac76b0ef-1c55-423b-b4a8-b649c0a50558	richers_20230924181032	35.00 bytes	00:02:08.167	00:02:08.215	No
2	70a3b05f-e765-407e-bd0c-1912a5148527	richers_20230924181432	35.00 bytes	00:01:49.168	00:01:49.205	No
3	8082703e-baaf-4f8a-ba7c-00309e7f8c1	richers_20230924181432	35.00 bytes	00:01:29.165	00:01:29.202	No
4	27cd9a6b-26a2-4a69-a57a-18a773c1344b	richers_20230924181432	35.00 bytes	00:01:08.164	00:01:08.169	No
5	7c24c0c8-8f6a-443b-b3ab-a888757f5aee	richers_20230924181032	35.00 bytes	00:00:49.172	00:00:49.195	No
6	ebab6644-2a0a-4cc0-aac8-b8a97517a5b5	richers_20230924181032	35.00 bytes	00:00:29.162	00:00:29.167	No
7	1865aa6f-1ab4-4290-b081-9523c078a4f8	richers_20230924181032	35.00 bytes	00:00:08.148	00:00:08.160	No

Vamos a parar el flujo y limpiar las colas para comenzar con el siguiente:

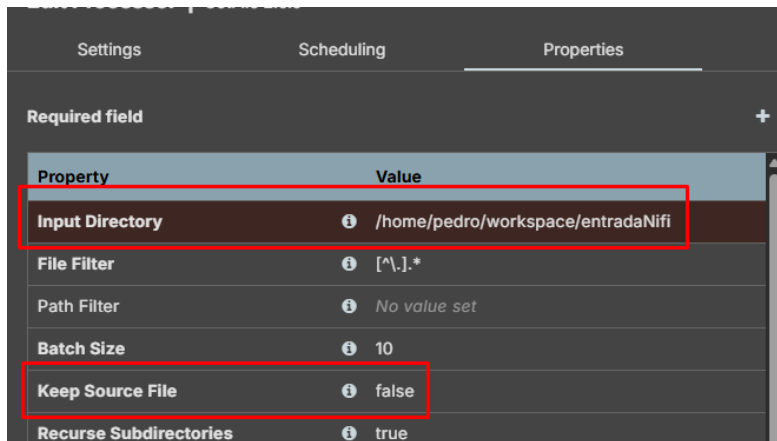


El segundo flujo consistirá en mover los archivos del directorio de entrada a uno de salida. Para ello, creamos un proceso **GetFile**:

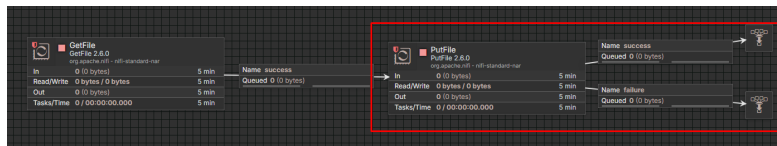




En sus propiedades establecemos el directorio de entrada:



Luego añadimos un proceso **PutFile**, indicando el directorio de salida, conectamos ambos procesos y añadimos **Funnels** para controlar los flujos de éxito y fallo. El flujo final debe quedar de la siguiente manera:



Al ejecutarlo, los ficheros se mueven del directorio de entrada al de salida:

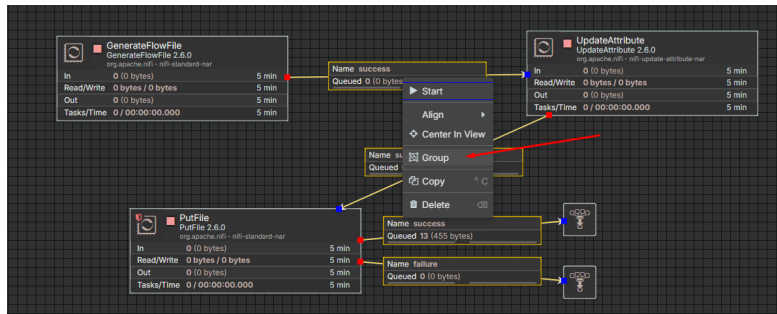
```

pedro@bda:~/workspace$ ll entradaNifi/
total 8
drwxrwxr-x 2 pedro pedro 4096 sep 24 16:22 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
pedro@bda:~/workspace$ ll salidaNifi/
total 56
drwxrwxr-x 2 pedro pedro 4096 sep 24 16:22 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161352
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161412
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161432
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161452
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161512
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161532
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161552
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161612
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161632
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162212
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162232
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162252
  
```

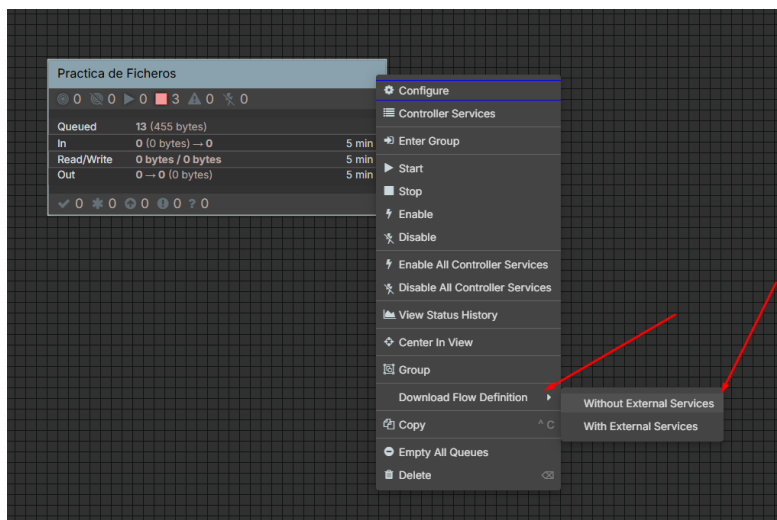
## 4. Uso de plantillas en NiFi

En este apartado vamos a aprender a crear y usar **plantillas** en NiFi, reutilizando uno de los flujos creados anteriormente.

Seleccionamos todo el flujo y, con clic derecho, escogemos la opción de crear un grupo:



Creamos el grupo, le ponemos el nombre que queramos y veremos cómo el flujo se representa ahora como un único elemento. Con clic derecho sobre él, seleccionamos la opción de descarga:



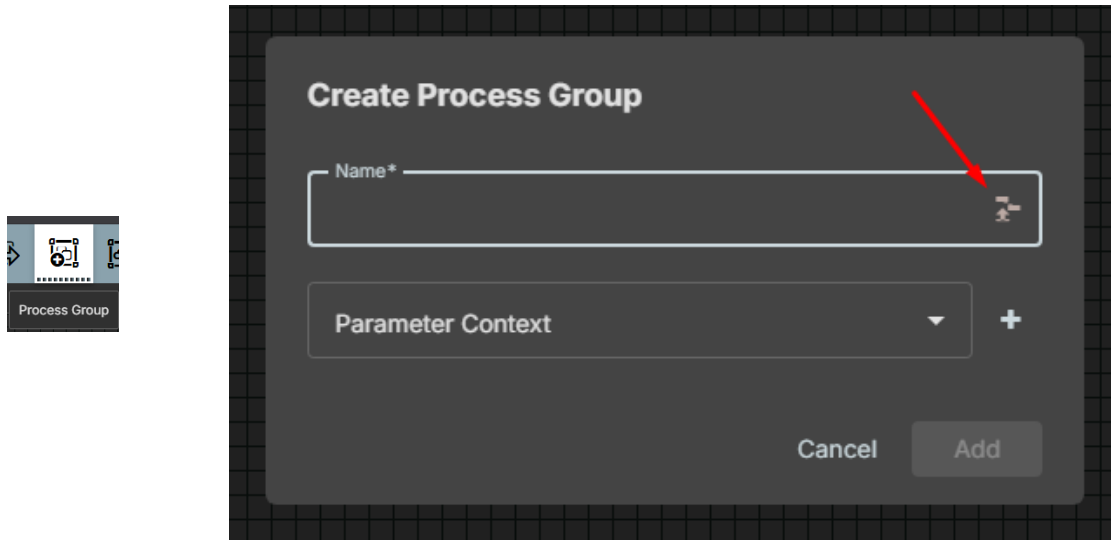
Esto nos descargará un fichero .json que contiene el grupo creado:

```

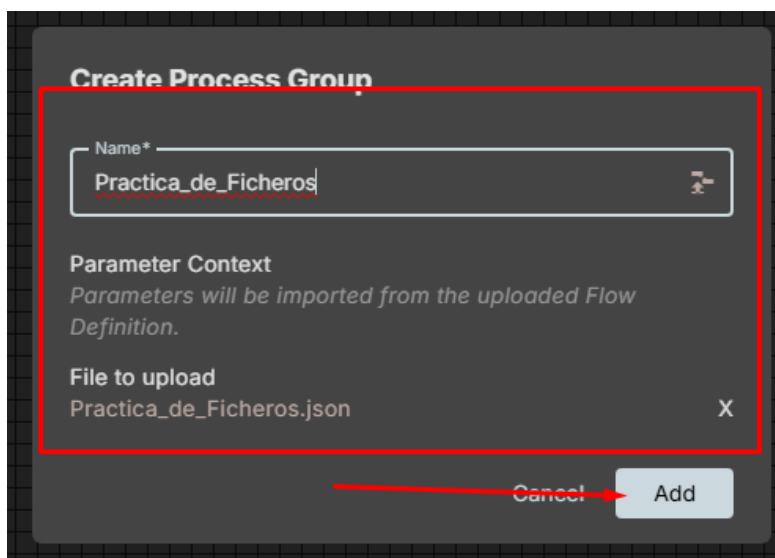
1  {
2    "flowContents": {
3      "identifier": "7c7c1e1-46ea-1bde-9baf-91cd43d1c2d1",
4      "instanceIdentifier": "7c7c1e1-46ea-1bde-9baf-91cd43d1c2d1",
5      "name": "Practica de Ficheros",
6      "comments": "",
7      "position": { "x": -904.0, "y": -489.0 },
8      "processGroups": [],
9      "remoteProcessGroups": [],
10     "processors": [
11       {
12         "identifier": "a027c02-1c1b-3f4b-8ec3-8809f782ab4a",
13         "instanceIdentifier": "a027c02-1c1b-3f4b-8ec3-8809f782ab4a",
14         "name": "UpdateAttribute",
15         "comments": "",
16         "position": { "x": -184.0, "y": -489.0 },
17         "type": "org.apache.nifi.processors.attributes.UpdateAttribute",
18         "bundle": { "group": "org.apache.nifi", "artifact": "nifi-update-attribute-nar", "version": "2.6.0" },
19         "properties": {
20           "Delete Attributes Expression": null,
21           "filename": "Ficheros_${now():format('yyyyMMdHms')}\"",
22           "Store State": "Do not store state",
23           "canonical-value-lookup-cache-size": "100",
24           "strictly-variables-initial-value": null
25         },
26         "propertyDescriptors": [
27           {
28             "name": "Delete Attributes Expression",
29             "displayName": "Delete Attributes Expression",
30             "identifier": "Delete Attributes Expression",
31             "sensitive": false,
32             "dynamic": false
33           },
34           {
35             "name": "filename",
36             "displayName": "filename",
37             "identifier": "filename",
38             "sensitive": false,
39             "dynamic": true
40           }
41         ]
42       }
43     ]
44   }
45 }

```

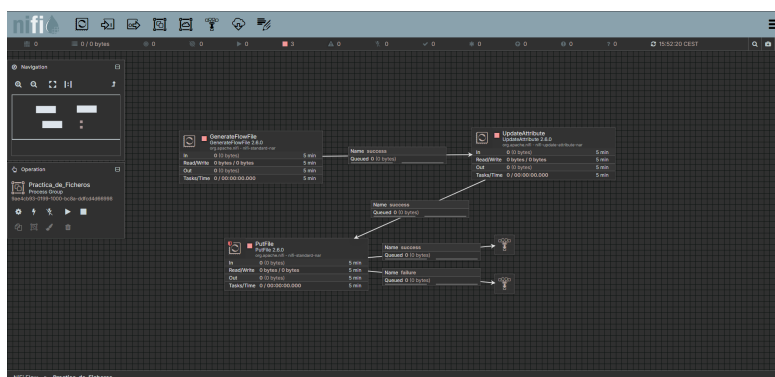
A continuación, borramos todo el flujo y dejamos el área de trabajo vacía. Una vez hecho, añadimos un **Process Group** y, en la ventana que aparece, seleccionamos la opción de importar un archivo:



Seleccionamos el archivo .json creado y lo añadimos:



Como podemos ver en el flujo, tenemos de vuelta el grupo creado anteriormente. Si damos doble clic, podemos entrar y ver el flujo original:



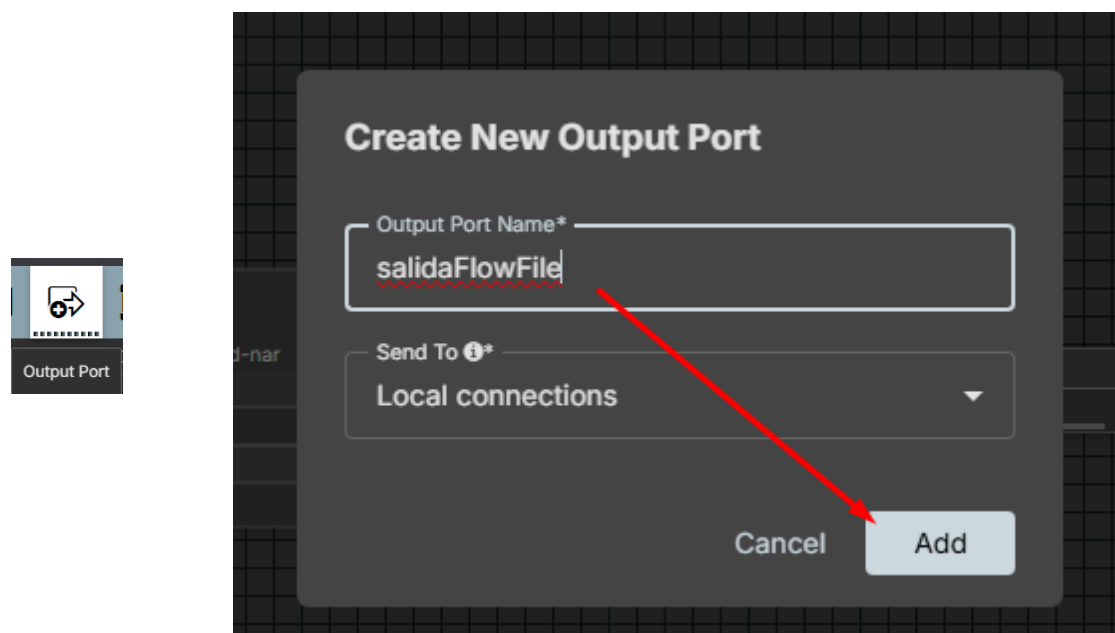
## 5. Practicando con grupos

Ahora que sabemos crear grupos, vamos a usarlos para crear flujos. Vamos borrar todo lo que teníamos hasta ahora y empezar de 0.

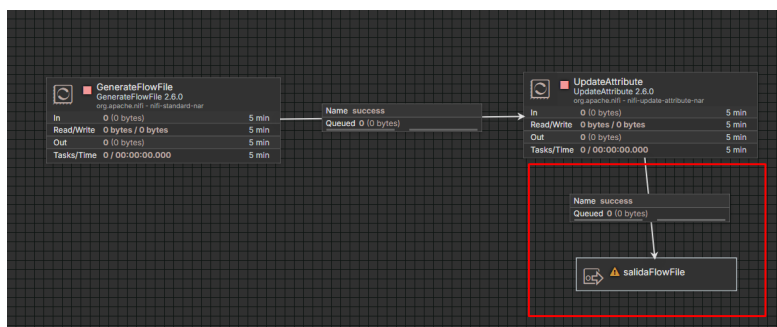
Primero vamos a crear un flujo que genere y renombre los archivos, y lo agrupamos:



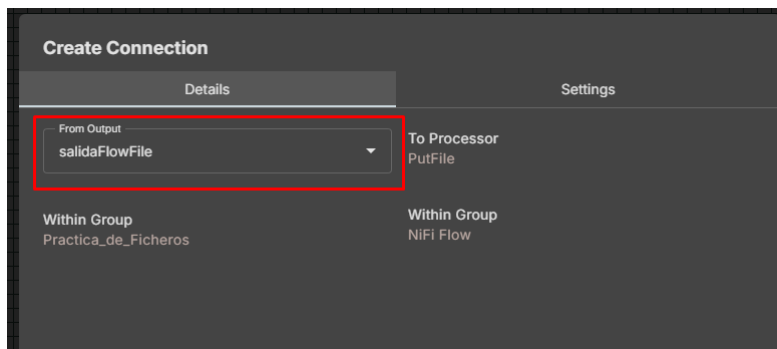
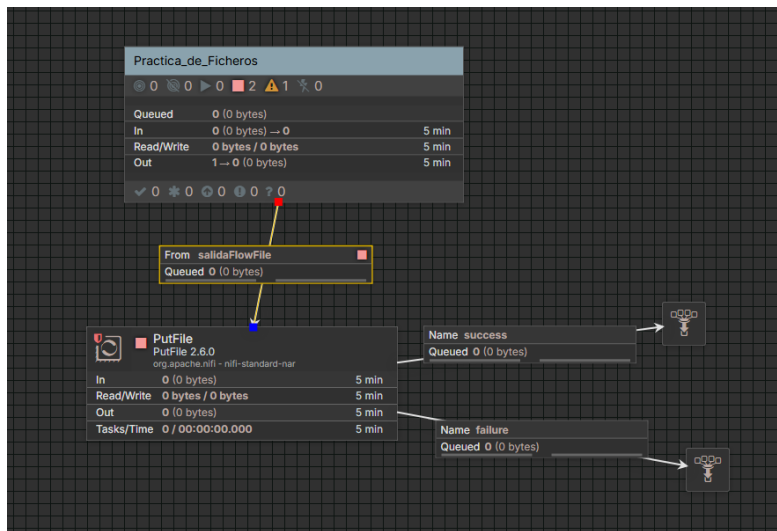
Entramos con doble clic al grupo y añadimos un **Output Port**:



El flujo final debería quedar tal que así:



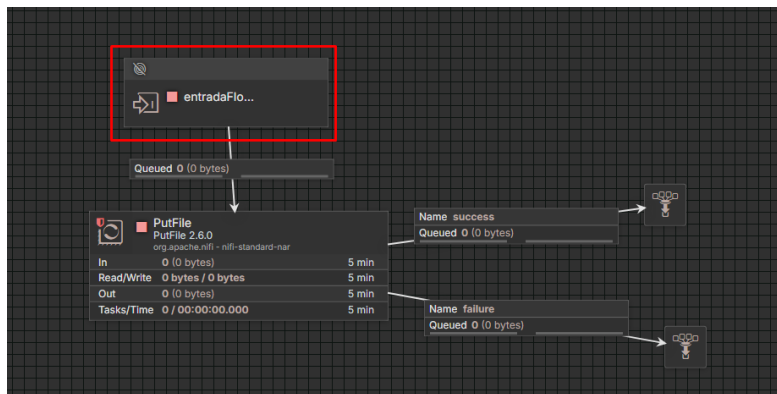
Ahora salimos del grupo y vamos a enlazar la salida con un flujo que coloque los archivos en un directorio:



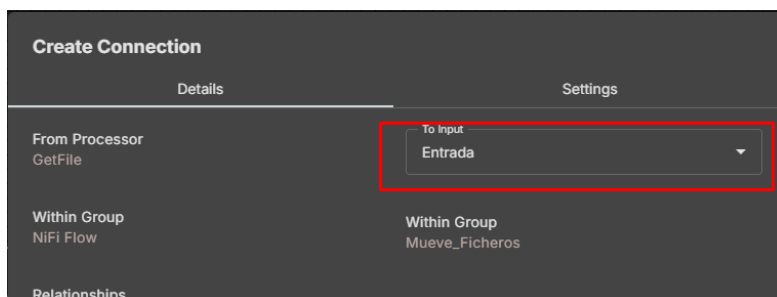
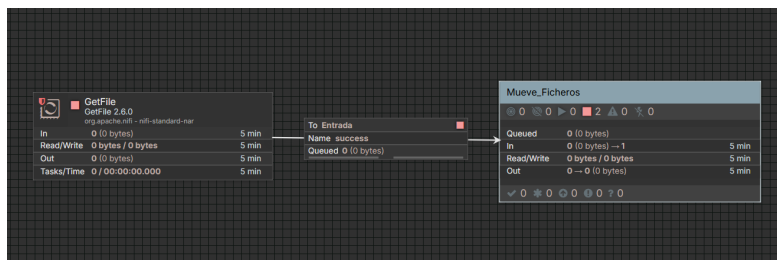
Finalmente, vamos a arrancar todo el flujo y comprobar el resultado:

```
pedro@bda:~/workspace/entradaNifi$ ll
total 40
drwxrwxr-x 2 pedro pedro 4096 sep 30 16:12 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:11 ficheros_20250930161143
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161203
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161217
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161222
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161227
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161232
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161237
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:12 ficheros_20250930161242
```

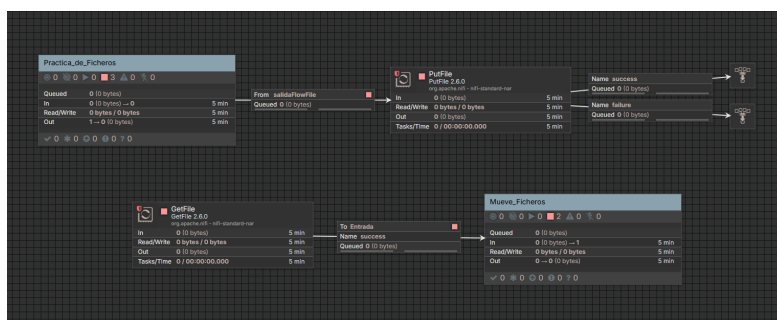
Ahora vamos a crear otro flujo que mueva los archivos a un directorio, y lo agrupamos. A este flujo le vamos a poner un **Input Port**:



Salimos del grupo, creamos un proceso para obtener ficheros y lo enlazamos al grupo, seleccionando de entrada el puerto que hemos añadido:



El resultado final de todo el flujo debería ser el siguiente:



Si lo probamos, veremos que los ficheros se crean en la entrada pero luego se mueven a la salida:

```
pedro@bda:~/workspace/entradaNifi$ ll
total 8
drwxrwxr-x 2 pedro pedro 4096 sep 30 16:20 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
pedro@bda:~/workspace/entradaNifi$ cd ../salidaNifi/
pedro@bda:~/workspace/salidaNifi$ ll
total 104
drwxrwxr-x 2 pedro pedro 4096 sep 30 16:20 ./
drwxrwxr-x 4 pedro pedro 4096 sep 24 16:10 ../
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161352
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161412
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161432
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161452
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161512
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161532
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161552
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161612
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924161632
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162212
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162232
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:22 ficheros_20250924162252
-rw-rw-r-- 1 pedro pedro 35 sep 24 16:23 ficheros_20250924162312
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161143
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161203
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161217
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161222
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161227
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161232
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161237
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930161242
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930162038
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930162043
-rw-rw-r-- 1 pedro pedro 35 sep 30 16:20 ficheros_20250930162048
```