Apache NiFi, paso a paso

stratebi
open business intelligence

1. INTRODUCCIÓN

Apache NiFi es un sistema de flujo de datos basado en los conceptos de programación basada en flujo (FBP, del inglés Flow-Based Programming).

Admite gráficos dirigidos de enrutamiento de datos, transformación y lógica de mediación del sistema. Apache NiFi tiene una interfaz de usuario web para el diseño, control, retroalimentación y monitorización de flujos de datos. Permite configurar la calidad del servicio, como tolerancia a pérdidas versus entrega garantizada, baja latencia versus alto rendimiento y colas basadas en prioridades.

Apache NiFi proporciona la procedencia de datos detallados para todos los datos recibidos, bifurcados, unidos, clonados, modificados, enviados y, en última instancia, abandonados al alcanzar su estado final configurado.

2. INSTALACIÓN

Tiene dos métodos de instalación: binarios y docker.

1. Binarios

Apache NiFi requiere Java 8 o 11. Se puede comprobar la versión de Java empleando el siguiente mandato:

```
java -version
```

Descargamos el archivo desde la página web de descargas y lo descomprimimos.

Para ejecutarlo en Windows basta con ir a la carpeta *bin* y hacer doble click sobre *run-nifi.bat*.

En Linux o MacOS se ejecuta empleando el siguiente comando:

```
bin/nifi.sh run
```

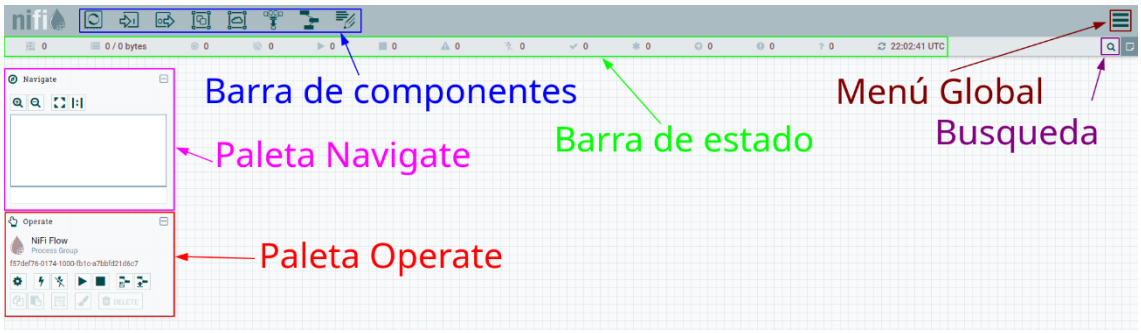
2. Docker

Se puede ejecutar sobre docker empleando el comando:

```
docker run --name nifi \
  -p 8080:8080 \
  -d \
  apache/nifi:latest
```

3. INTERFAZ WEB

Podemos acceder a la interfaz web de Apache NiFi navegando http://localhost:8080/nifi/. Por defecto no hay permisos configurados, por lo que cualquiera puede ver y modificar el flujo de datos.



- Barra de componentes - consiste en los elementos que se pueden arrastrar al lienzo para construir el flujo de datos.
- Barra de estado - proporciona información sobre la cantidad de subprocesos que están actualmente activos en el flujo, la cantidad de datos que existen actualmente en el flujo, Grupos de Procesos Remotos (Remote Process Groups) en cada estado (transmitiendo, no transmitiendo), Procesadores (Processors) en cada estado (detenido, en ejecución, no válido, deshabilitado), Grupos de Procesos versionados en cada estado (actualizado, modificado localmente, obsoleto, modificado y obsoleto localmente, fallo de sincronización) y la marca de tiempo en la que se actualizó por última vez toda esta información.
- Paleta Operate - consiste en botones que se utilizan para administrar el flujo, el acceso de los usuarios y las propiedades del sistema.
- Paleta Navigate - permite navegar por el lienzo.
- Menú Global - contiene opciones que permiten manipular componentes existentes en el lienzo.
- Busqueda - permite buscar componentes en el lienzo.

4. CONCEPTOS

1. DataFlow Manager (DFM)

Es un usuario de NiFi que tiene permisos para agregar, eliminar y modificar componentes de un flujo de datos.

2. FlowFile

Es un registro de datos, que consiste en un puntero a su contenido y atributos para respaldar el contenido, que está asociado con uno o más eventos de procedencia. El contenido son los datos representados por FlowFile. Los atributos son características que brindan información o contexto sobre los datos; están formados por pares clave-valor. Todos los FlowFiles tienen los siguientes atributos estándar:

- **uuid** - un identificador único que distingue el FlowFile de otros FlowFiles del sistema.
- **filename** - un nombre de archivo legible que se puede usar al almacenar los datos en el disco o en un servicio externo.
- **path** - la ruta que se usa para almacenar datos en un disco o en un servicio externo.

3. Procesador (Processor)

Es el componente NiFi que se utiliza para escuchar, extraer datos de fuentes externas, publicar datos en fuentes externas, y enrutar, transformar o extraer información de FlowFiles. Apache NiFi proporciona varios tipos de procesadores, por ejemplo:

- **Transformación de datos**
    - **CompressContent** – comprimir y descomprimir contenido.
    - **ConvertCharacterSet** – convierte los caracteres a otra codificación.
    - **EncryptContent** – cifrar o descifrar contenido.
    - **ReplaceText** – modificar un texto usando expresiones regulares.
    - **TransformXml** – aplicar transformaciones XSLT al contenido del XML.

- **Enrutamiento y mediación**
    - ControlRate – regula la velocidad a la que los datos pueden fluir a través de una parte del flujo.
    - DetectDuplicate – detectar FlowFiles duplicados según los criterios definidos por el usuario.
    - DistributeLoad – balanceador de carga.
    - MonitorActivity – envía una notificación cuando transcurre un período de tiempo definido por el usuario sin que llegue ningún dato a través de un punto particular del flujo.
    - RouteOnAttribute – enruta un FlowFile según sus atributos.
    - ScanAttribute: escanea el conjunto de atributos definidos por el usuario en un FlowFile, verificando si alguno de los atributos coincide con los términos encontrados en un diccionario definido por el usuario.
    - RouteOnContent – busca en el contenido de un FlowFile para ver si coincide con alguna expresión regular definida por el usuario; si es así, el FlowFile se enruta a la relación configurada.
    - ScanContent – busca en el contenido de un FlowFile los términos que están en un diccionario definido por el usuario y enruta basándose en la presencia o ausencia de esos términos.
    - ValidateXml – validación de contenido XML contra un esquema XML; enruta FlowFile en función de si el contenido del FlowFile es válido o no de acuerdo con el esquema XML definido por el usuario.

- **Acceso a bases de datos**
    - ConvertJSONToSQL – convierte un documento JSON en una sentencia SQL INSERT o UPDATE que luego se puede pasar al procesador PutSQL.
    - ExecuteSQL – ejecuta un comando SQL SELECT definido por el usuario, escribiendo los resultados en un FlowFile en formato Avro.
    - PutSQL – actualiza una base de datos mediante la ejecución de la declaración SQL DDM definida por el contenido del FlowFile.

- **Extracción de atributos**
    - **EvaluateJsonPath** – expresiones JSONPath que se evalúan contra el contenido JSON para reemplazar el contenido del FlowFile o extraer el valor en un atributo nombrado por el usuario.
    - **EvaluateXPath** – expresiones XPath que se evalúan contra el contenido XML para reemplazar el contenido del FlowFile o extraer el valor en un atributo nombrado por el usuario.
    - EvaluateXQuery – consulta XQuery que se evalúa contra el contenido XML para reemplazar el contenido del FlowFile o extraer el valor en un atributo nombrado por el usuario.
    - ExtractText – expresiones regulares que se evalúan contra el contenido textual del FlowFile, los valores que se extraen se añaden como atributos.
    - HashAttribute – aplica una función hash contra la concatenación de una lista de atributos existentes definida por el usuario.
    - HashContent – aplica una función hash contra el contenido de un FlowFile y añade el valor hash como un atributo.
    - IdentifyMimeType – evalúa el contenido de un FlowFile para determinar qué tipo de archivo encapsula el FlowFile. Este procesador es capaz de detectar muchos tipos de MIME diferentes, como imágenes, documentos de procesador de texto, texto y formatos de compresión, solo por nombrar algunos.
    - UpdateAttribute – añade o actualiza cualquier cantidad de atributos definidos por el usuario a un FlowFile.
- **Interacción con el sistema**
    - **ExecuteProcess** – ejecuta un comando / proceso; la salida estándar del proceso se redirige de modo que el contenido que se escribe en la salida estándar se convierte en el contenido del FlowFile de salida; este procesador es un procesador de origen: se espera que su salida genere un nuevo FlowFile, y se espera que la llamada al sistema no reciba ninguna entrada.
    - ExecuteStreamCommand – ejecuta un comando / proceso; el contenido del FlowFile se transmite opcionalmente a la entrada estándar del proceso; el contenido que se escribe en la salida estándar se convierte en el contenido del FlowFile de salida. Este procesador no se puede utilizar como un procesador de origen; se debe alimentar con FlowFiles de entrada para realizar su trabajo.

- **Ingestión de datos**
    - **GetFile** – transmite el contenido de un archivo desde un disco local a NiFi y luego elimina el archivo original; se espera que este procesador mueva el archivo de una ubicación a otra ubicación y no se debe utilizar para copiar los datos.
    - **GetSFTP** – descarga el contenido de un archivo remoto a través de SFTP y luego elimina el archivo original; se espera que este procesador mueva los datos de una ubicación a otra ubicación y no se debe utilizar para copiar los datos.
    - **GetHTTP** – descarga el contenido de una URL remota basada en HTTP o HTTPS; el procesador recordará la ETag y la fecha de última modificación para garantizar que los datos no se ingieran continuamente.
    - **FetchS3Object** – recupera el contenido de un objeto de S3 de AWS; el FlowFile de salida contiene el contenido recibido de S3.
    - **GetKafka** – recupera mensajes de Apache Kafka; los mensajes se pueden emitir como un FlowFile por mensaje o se pueden agrupar mediante un delimitador.
    - **GetTwitter** – permite escuchar Tittwer, creando un FlowFile para cada tweet que se recibe.
    - GetHDFS – supervisa un directorio especificado por el usuario en HDFS; siempre que un nuevo archivo ingresa a HDFS, se copia en NiFi y se elimina de HDFS; se espera que este procesador mueva el archivo de una ubicación a otra ubicación y no se debe utilizar para copiar los datos.
- **Salida de datos / Envío de datos**
    - PutEmail – envía un correo electrónico a los destinatarios configurados; el contenido del FlowFile se envía opcionalmente como un archivo adjunto.
    - PutFile – escribe el contenido de un FlowFile en un directorio del sistema de archivos local.
    - PutSFTP – copia el contenido de un FlowFile en un servidor SFTP remoto.
    - PutKafka – envía el contenido de un FlowFile como mensaje a Apache Kafka; el FlowFile se puede enviar como un solo mensaje o como un delimitador.
    - PutSQL – visto anteriormente.

- **División y agregación**
    - **SplitText** – permite dividir en uno o más FlowFiles según el número configurado.
    - SplitJson – permite dividir un objeto JSON que se compone de un array o muchos objetos hijos en un FlowFile por elemento JSON.
    - SplitXml – permite dividir un mensaje XML en muchos FlowFiles, cada uno conteniendo un segmento del original.
    - UnpackContent – descomprime diferentes tipos de archivo, como ZIP y TAR; cada fichero dentro del archivo se transfiere luego como un solo FlowFile.
    - MergeContent – responsable de fusionar muchos FlowFiles en un solo FlowFile.
    - SegmentContent – segmenta un FlowFile en muchos FlowFiles más pequeños en función de algún tamaño de datos configurado.
    - SplitContent – divide un solo FlowFile en muchos FlowFiles, de manera similar a SegmentContent.

Estos son solo algunos de los componentes más usados, pero Apache NiFi proporciona más componentes.

4. Relacion (Relationship)

Cada Procesador tiene cero o más relaciones definidas. Estas relaciones se nombran para indicar el resultado de procesar un FlowFile. Una vez que un Procesador ha terminado de procesar un FlowFile, enrutará el FlowFile a una de las Relaciones. Luego, un DFM puede conectar cada una de estas relaciones con otros componentes para especificar dónde debe ir el FlowFile en cada resultado de procesamiento potencial.

5. Conexión (Connection)

Un flujo de datos automatizado se crea arrastrando componentes desde la Barra de Componentes al lienzo y luego conectando los componentes a través de Conexiones. Cada Conexión consta de una o más relaciones. Para cada conexión que se dibuja, un DFM puede determinar qué relaciones deben usarse para la conexión. Esto permite que los datos se enruten de diferentes maneras según el resultado del procesamiento. Cada conexión alberga una cola FlowFile. Cuando un FlowFile se transfiere a una Relación en particular, se agrega a la cola que pertenece a la Conexión asociada.

6. Controller Service

Los tiene que agregar y configurar un DFM en la interfaz de usuario. Se iniciarán cuando NiFi se inicie y proporcionarán información para que la utilicen otros componentes. La idea es que, en lugar de configurar esta información en cada Procesador, el Controller Service la proporcione para que cualquier Procesador la utilice según sea necesario.

7. Reporting Task

Se ejecutan en segundo plano para proporcionar informes estadísticos sobre lo que está sucediendo en la instancia de NiFi. El DFM agrega y configura Reporting Tasks en la interfaz de usuario.

8. Funnel

Es un componente de NiFi que se utiliza para combinar los datos de varias conexiones en una sola conexión.

9. Grupo de Procesos (Process Group)

Cuando un flujo de datos se vuelve complejo, es beneficioso razonar sobre el flujo de datos a un nivel más alto y abstracto. NiFi permite que varios componentes, como Procesadores, se agrupen en un Grupo de Procesos. Luego, en la interfaz de usuario de NiFi, un DFM puede conectar varios Grupos de Procesos en un flujo de datos lógico.

10. Puerto (Port)

Los flujos de datos que se construyen utilizando uno o más grupos de procesos necesitan una forma de conectar un Grupo de Procesos a otros componentes del flujo de datos. Esto se logra mediante el uso de puertos. Un DFM puede agregar cualquier cantidad de puertos de entrada y puertos de salida a un Grupo de Procesos y nombrar estos puertos de manera apropiada.

11. Grupo de Procesos Remotos (Remote Process Group)

A veces es necesario transferir datos de una instancia de NiFi a otra. Los Grupos de Procesos Remotos suelen ser la forma más fácil de transferir datos entre instancias de NiFi.

12. Bulletin

La interfaz de usuario de NiFi proporciona una cantidad significativa de monitorización y retroalimentación sobre el estado actual de la aplicación. Además de las estadísticas continuas y el estado actual proporcionado para cada componente, los componentes pueden informar *Boletines*. Siempre que un componente informa un boletín, se muestra un icono de boletín en ese componente. Los boletines a nivel del sistema se muestran en la barra de estado cerca de la parte superior de la página. El uso del mouse para pasar el cursor sobre ese ícono proporcionará información sobre herramientas que muestra la hora y la gravedad del boletín, así como el mensaje del boletín. Los boletines de todos los componentes también se pueden ver y filtrar en la página del tablero de anuncios, disponible en el menú global.

13. Template

Muchas veces, un flujo de datos se compone de muchos subflujos que podrían ser reutilizados. NiFi permite seleccionar una parte del flujo de datos y crear una plantilla. A esta plantilla se le da un nombre y luego se puede arrastrar al lienzo al igual que los otros componentes. Como resultado, varios componentes pueden ser combinados para formar un bloque de construcción más grande a partir del cual crear un flujo de datos. Estas plantillas también se pueden exportar como XML e importar a otra instancia de NiFi.

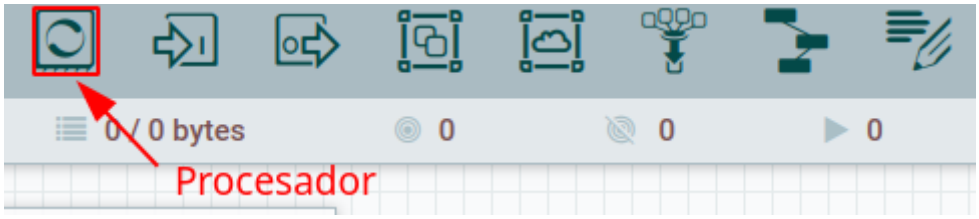
5. CREACIÓN DE DATAFLOW

Vamos a implementar un flujo para descargar los datos Covid-19 por Municipios y Distritos de Madrid usando la API publica de la comunidad. Esta API devuelve un JSON con el siguiente formato:

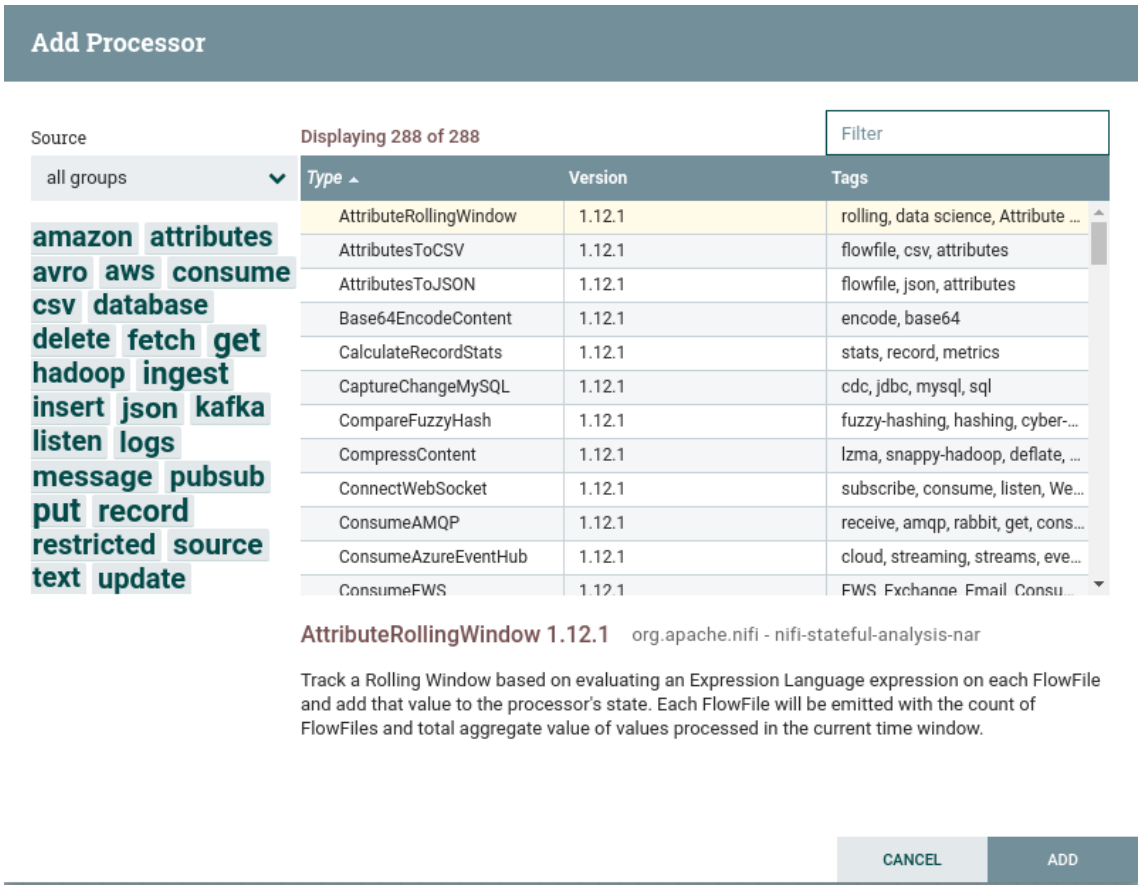
```
{
  "data": [
    {
      "municipio_distrito": "Madrid-Retiro",
      "codigo_geometria": "079603",
      "tasa_incidencia_acumulada_activos_ultimos_14dias": 212.040195445745,
      "tasa_incidencia_acumulada_ultimos_14dias": 624.387136786879,
      "tasa_incidencia_acumulada_total": 3302.12794488631,
      "casos_confirmados_totales": 3940,
      "casos_confirmados_activos_ultimos_14dias": 253,
      "casos_confirmados_ultimos_14dias": 745,
      "fecha_informe": "2020/09/29 11:15:00"
    },
    ........
}
```

Vamos a crear un fichero CSV por cada Municipio o Distrito (usando el campo *municipio_ distrito*).

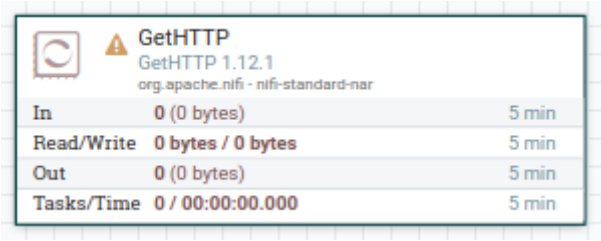
Comenzamos a crear el flujo de datos arrastrando un procesador al lienzo.



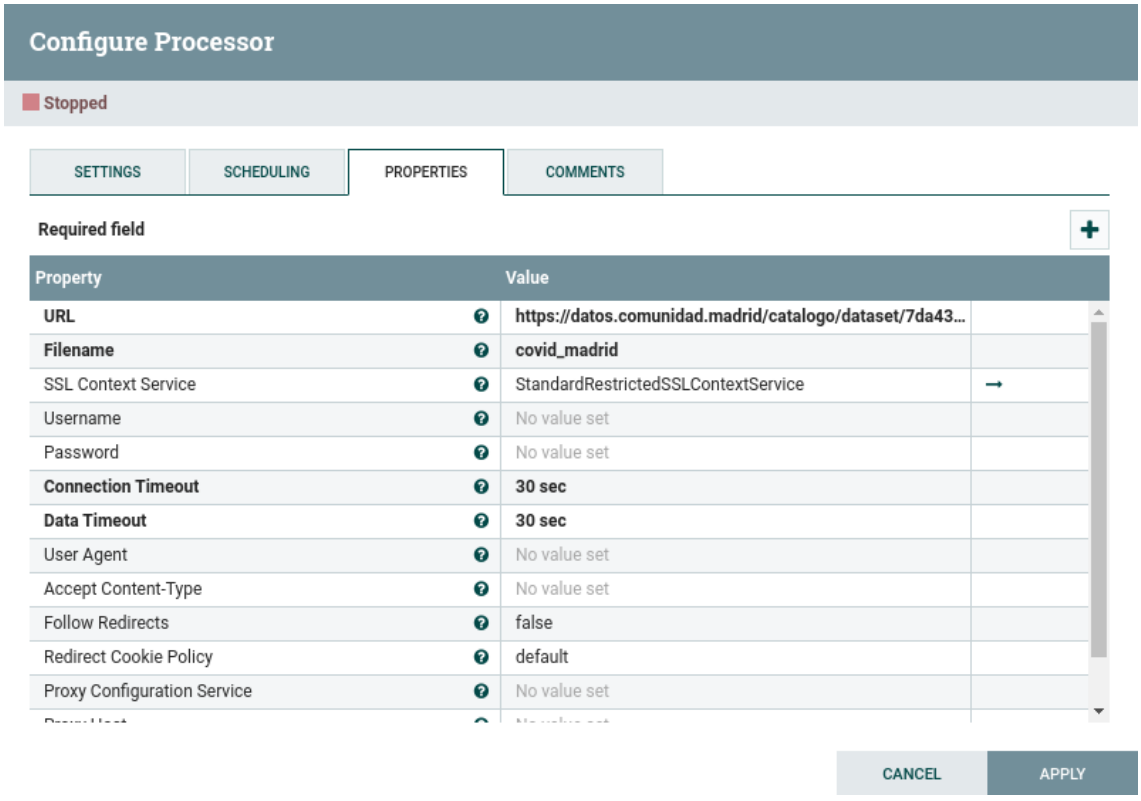
Esto nos proporciona un dialogo que nos permite elegir el Procesador.



Buscamos el Procesador GetHTTP y hacemos click en *ADD*.

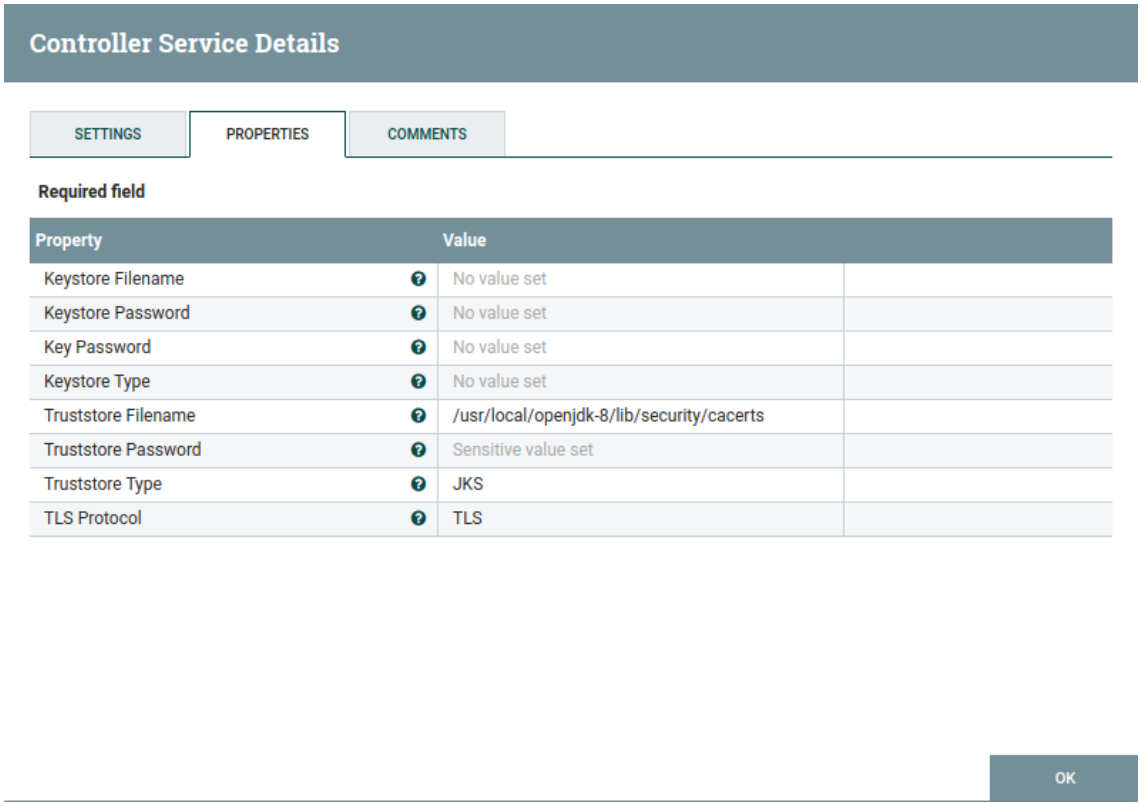


Podemos configurarlo haciendo click derecho sobre GetHTTP y eligiendo *Configure*.



Las propiedades que están disponibles dependen del tipo de procesador y generalmente son diferentes para cada tipo. Las propiedades que están en negrita son propiedades obligatorias. El procesador no se puede iniciar hasta que se hayan configurado todas las propiedades necesarias. Por tanto, tenemos que definir el URL especificando la API publica de Madrid y Filename que es el nombre del FlowFile dejando las demás propiedades con el valor por defecto.

A la API se accede mediante el protocolo HTTPS por lo que hay que definir la propiedad SSL Context Service. Usamos StandardRestrictedSSLContextService. A la derecha de este campo aparece una flecha que nos manda al menú de configuración de *Controller Services* donde tenemos que configurar el StandardRestrictedSSLContextService.

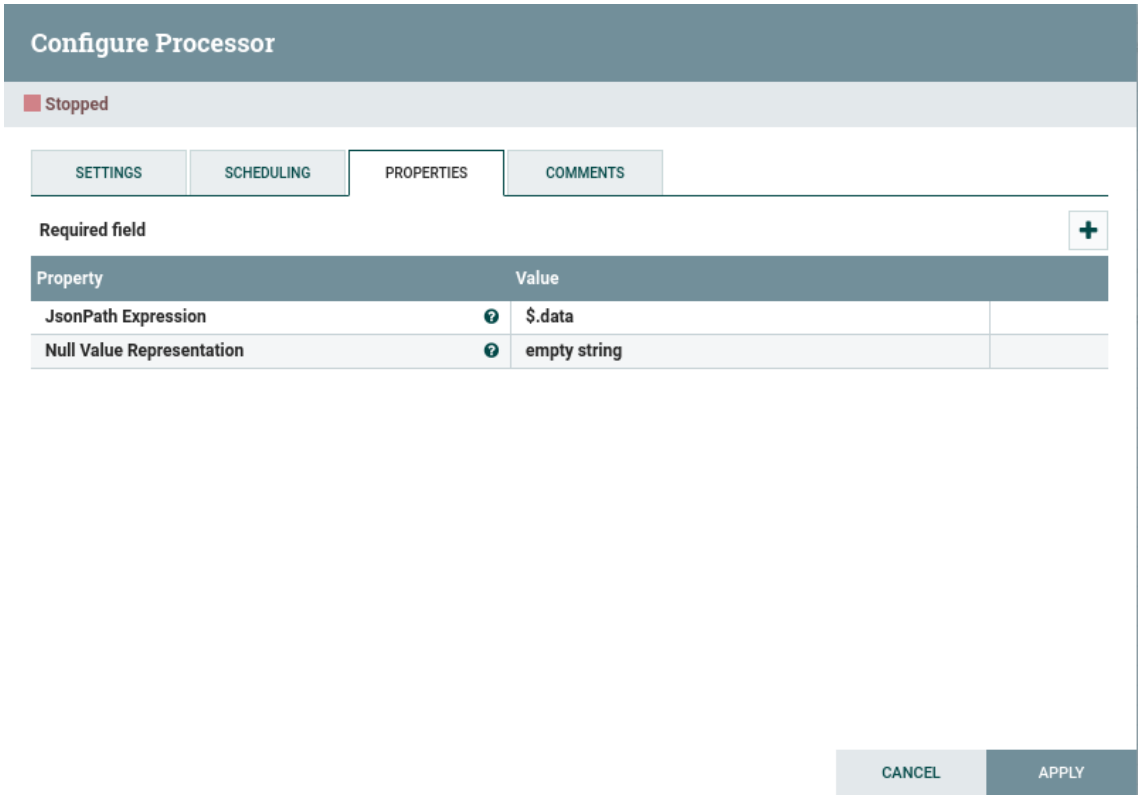


Usamos el Truststore de Java:

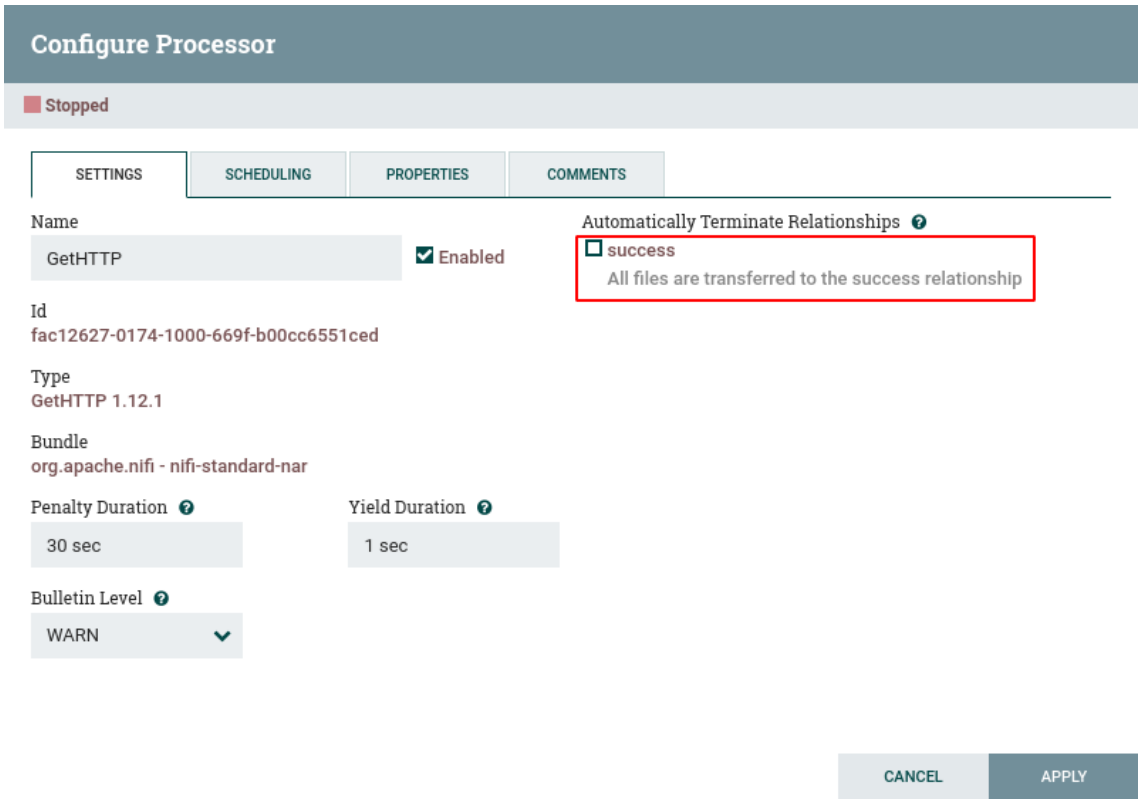
- Truststore Filename – el nombre del fichero del Truststore; se encuentra en *$JAVA_HOME/lib/security/cacerts*.
- Truststore Password – la contrasena del Trustore; por defecto *changeit*.
- Truststore Type – el tipo del Truststore, en este caso *JKS*.

Una vez configurado tenemos que habilitar el Controller Service haciendo click en el botón *rayo*.

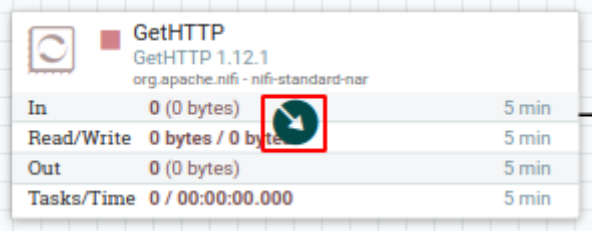
Arrastramos otro Procesador al lienzo y lo configuramos para que divida el array *data* del JSON en un FlowFile por elemento.



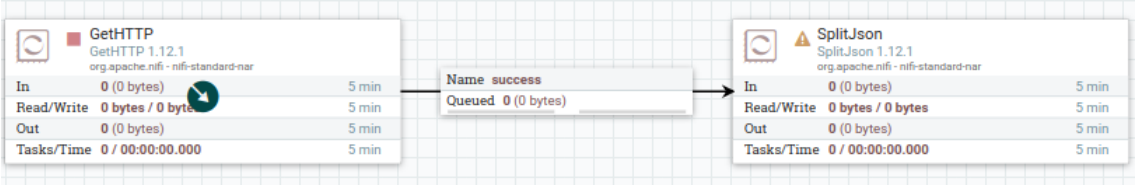
Cada Procesador tiene un conjunto de "Relaciones" definidas a las que puede enviar datos. Cuando un procesador termina de manejar un FlowFile, lo transfiere a una de estas relaciones. Esto permite al usuario configurar cómo manejar los FlowFiles en función del resultado del procesamiento. Por ejemplo, GetHTTP define la Relación success, SplitJSON define failure, original y split.



Podemos enviar la salida del procesador GetHTTP al procesador SplitJson. Pasamos el mouse sobre el Procesador GetFile con el mouse y aparecerá un ícono de conexión (Conexión) en el medio del Procesador.



Arrastramos este icono del procesador GetHTTP al procesador SplitJson. Esto nos muestra un diálogo para elegir qué relaciones queremos incluir para esta conexión. Debido a que GetHTTP tiene una sola relación, success, se selecciona automáticamente.



Añadimos un Procesador EvaluateJsonPath para extraer los campos del JSON en atributos. Podemos añadir atributos haciendo click en el botón +. Lo configuramos de la siguiente manera:

Configure Processor

Stopped

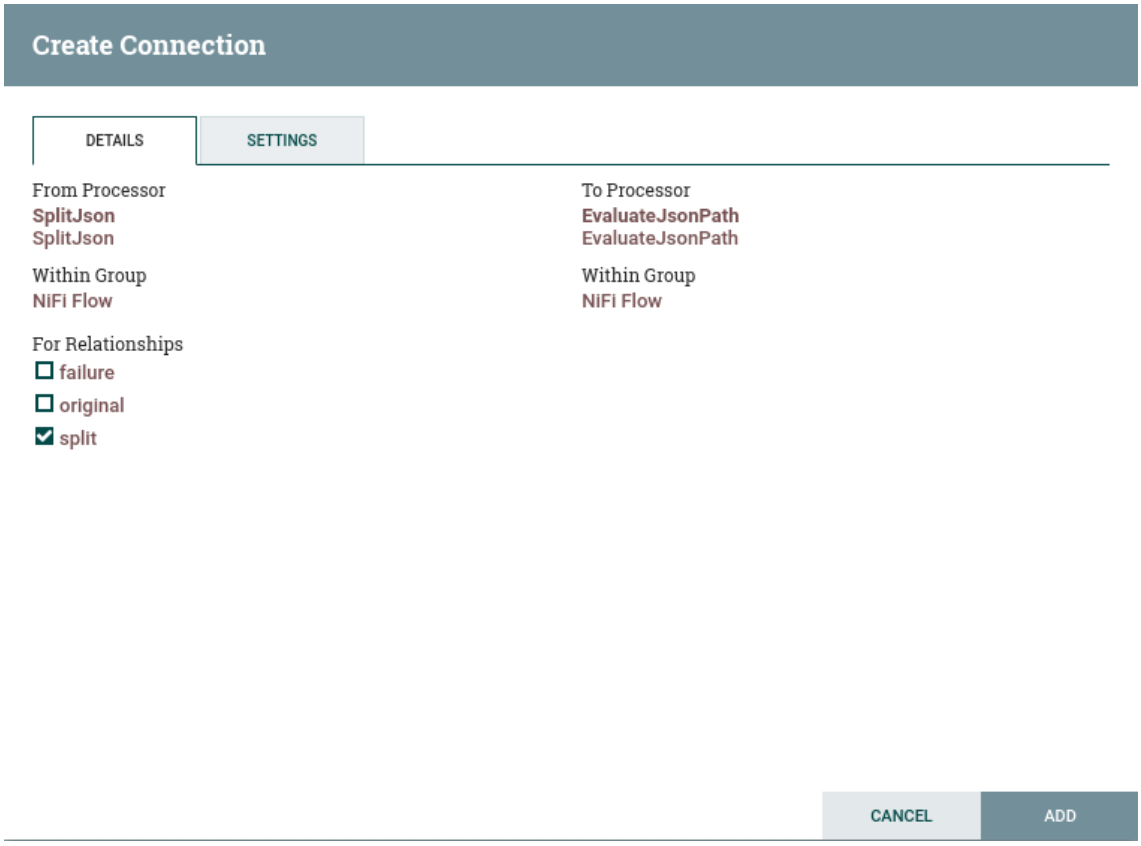
SETTINGS | SCHEDULING | PROPERTIES | COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
casos_confirmados_activos_ultimos_14dias	$.casos_confirmados_activos_ultimos_14dias
casos_confirmados_totales	$.casos_confirmados_totales
casos_confirmados_ultimos_14dias	$.casos_confirmados_ultimos_14dias
fecha_informe	$.fecha_informe
municipio_distrito	$.municipio_distrito
tasa_incidencia_acumulada_activos_ultimos_14dias	$.tasa_incidencia_acumulada_activos_ultimos_14dias
tasa_incidencia_acumulada_total	$.tasa_incidencia_acumulada_total
tasa_incidencia_acumulada_ultimos_14dias	$.tasa_incidencia_acumulada_ultimos_14dias

CANCEL | APPLY

Creamos una Conexión entre SplitJson y EvaluateJsonPath para la Relación split.



Arrastramos un Procesados ReplaceText al lienzo para cambiar el contenido del FlowFile de JSON a CSV. Cambiamos Replacement Strategy a Always Replace y juntantamos todos los atributos extraídos en el paso anterior usando coma (,) como delimitador. Al final de la línea tenemos que insertar una línea nueva, esto se hace con la combinación de teclas *Ctrl* + *Enter*.

```
"${municipio_distrito}",${tasa_incidencia_acumulada_activos_ultimos_14dias},$
{tasa_incidencia_acumulada_ultimos_14dias},${tasa_incidencia_acumulada_total}
,${casos_confirmados_totales},${casos_confirmados_activos_ultimos_14dias},${c
asos_confirmados_ultimos_14dias},"${fecha_informe}"
```

Configure Processor

Stopped

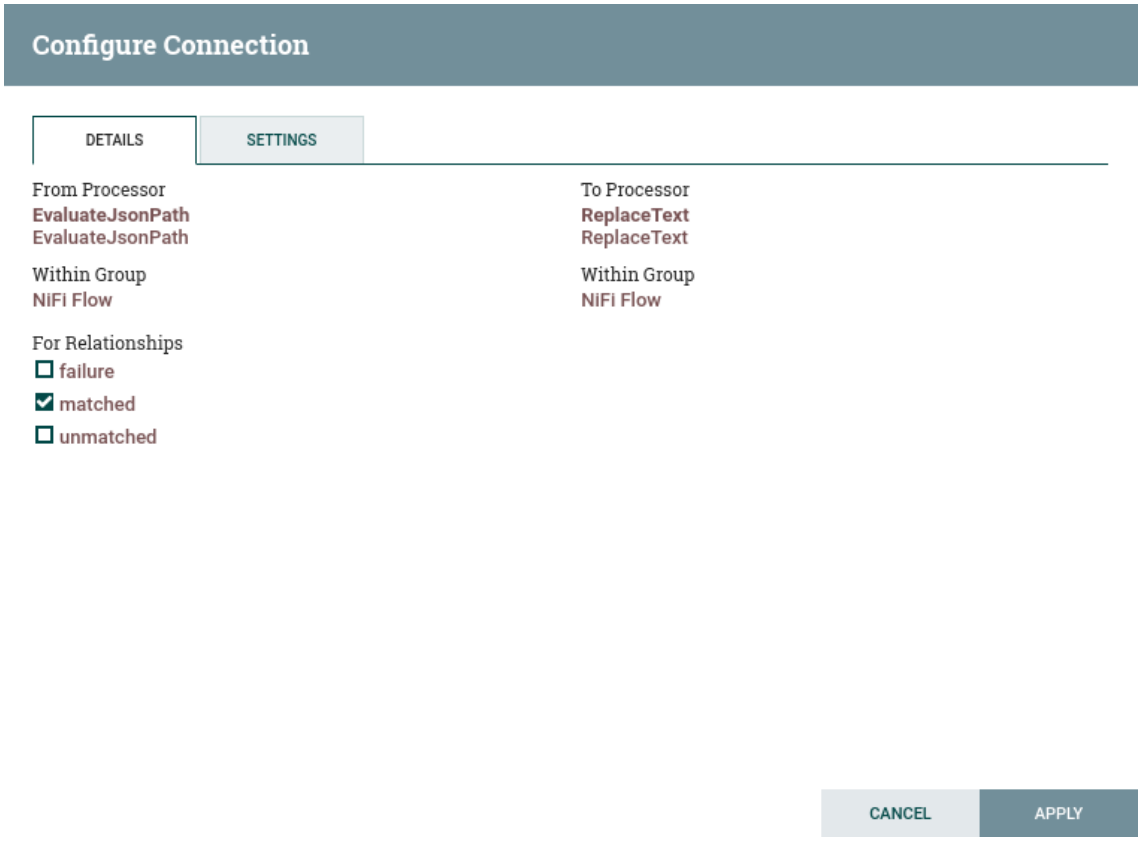
SETTINGS | SCHEDULING | PROPERTIES | COMMENTS

Required field

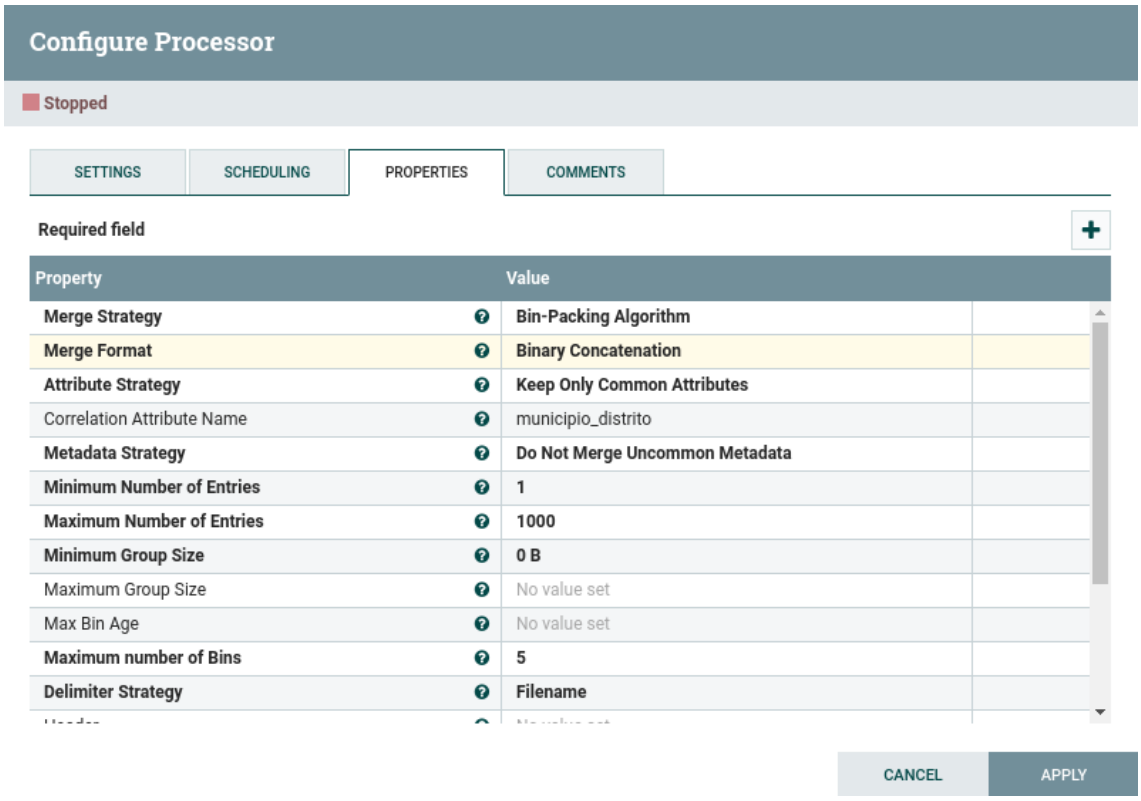
Property	Value
Search Value	(?s)(^.*$)
Replacement Value	"${municipio_distrito}",${tasa_incidencia_acumulada_activ...
Character Set	UTF-8
Maximum Buffer Size	1 MB
Replacement Strategy	Always Replace
Evaluation Mode	Entire text
Line-by-Line Evaluation Mode	All

CANCEL APPLY

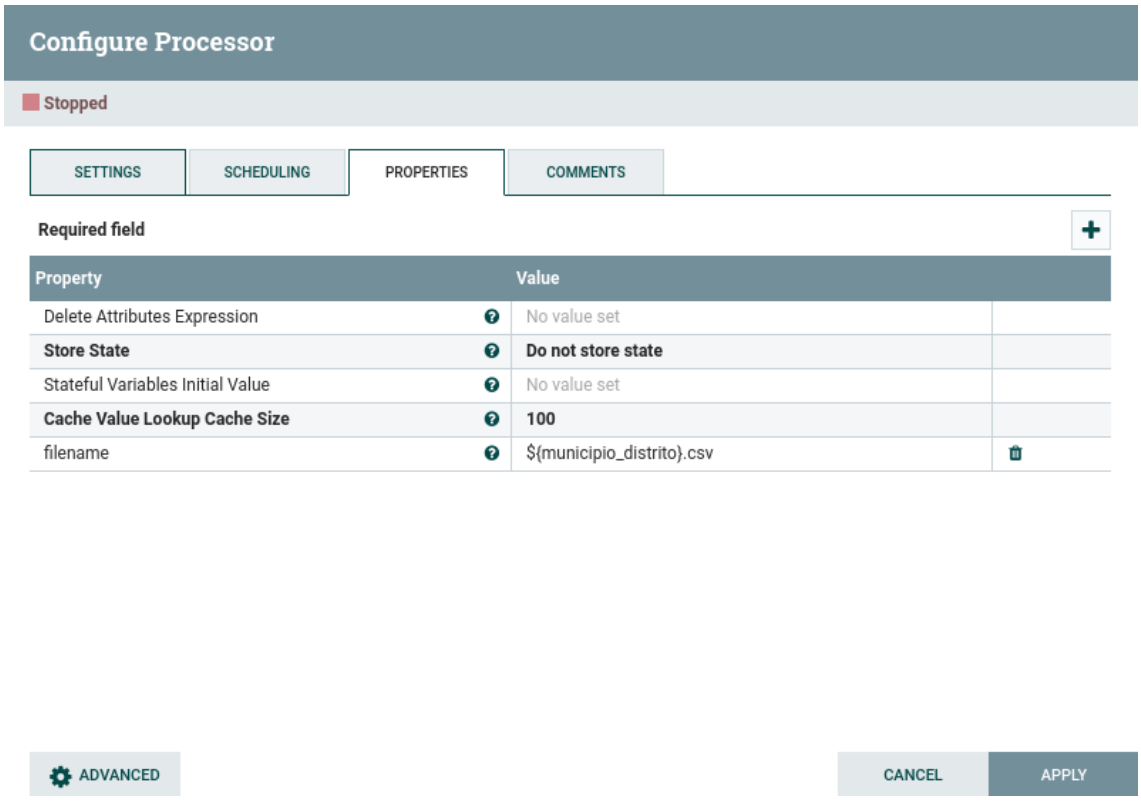
Creamos una Conexión entre EvaluateJsonPath y ReplaceText para la Relación matched.



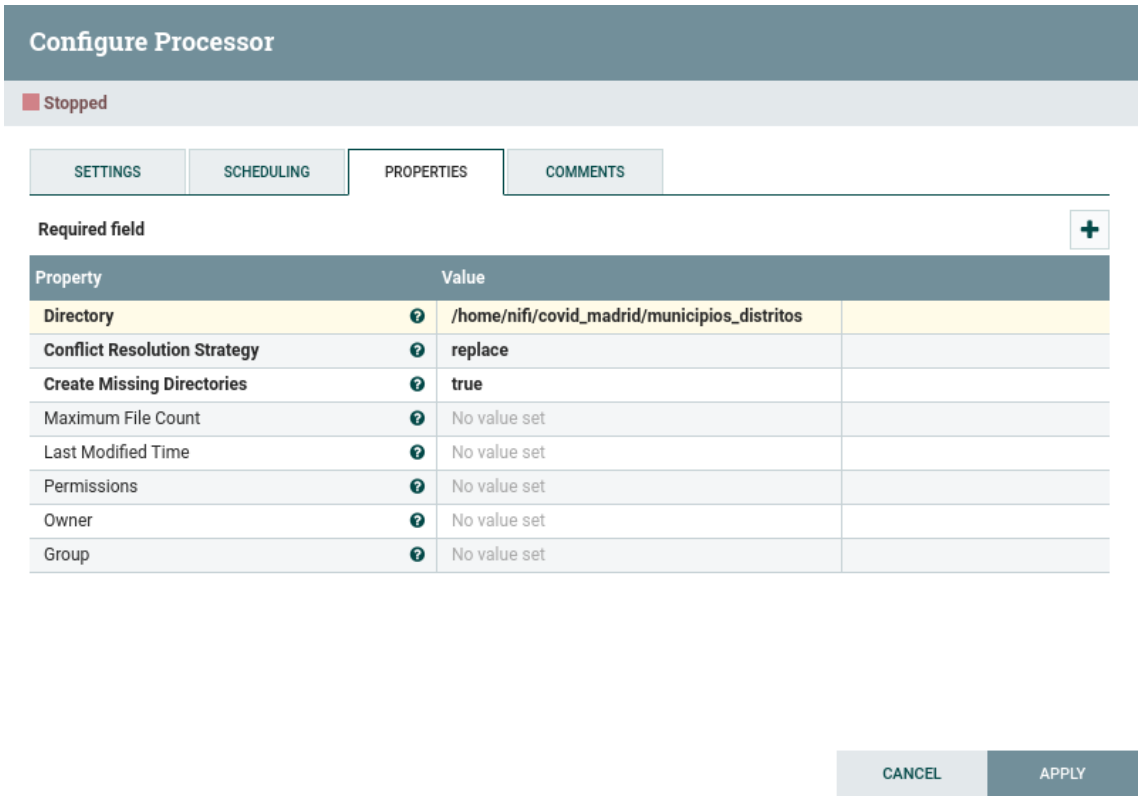
Añadimos un Procesador MergeContent para unir los datos por el campo municipio_distrito. Luego, creamos una Conexión entre ReplaceText y MergeContent para la Relación success. Configuramos el Procesador para que combine los FlowFiles por el campo municipio_distrito.



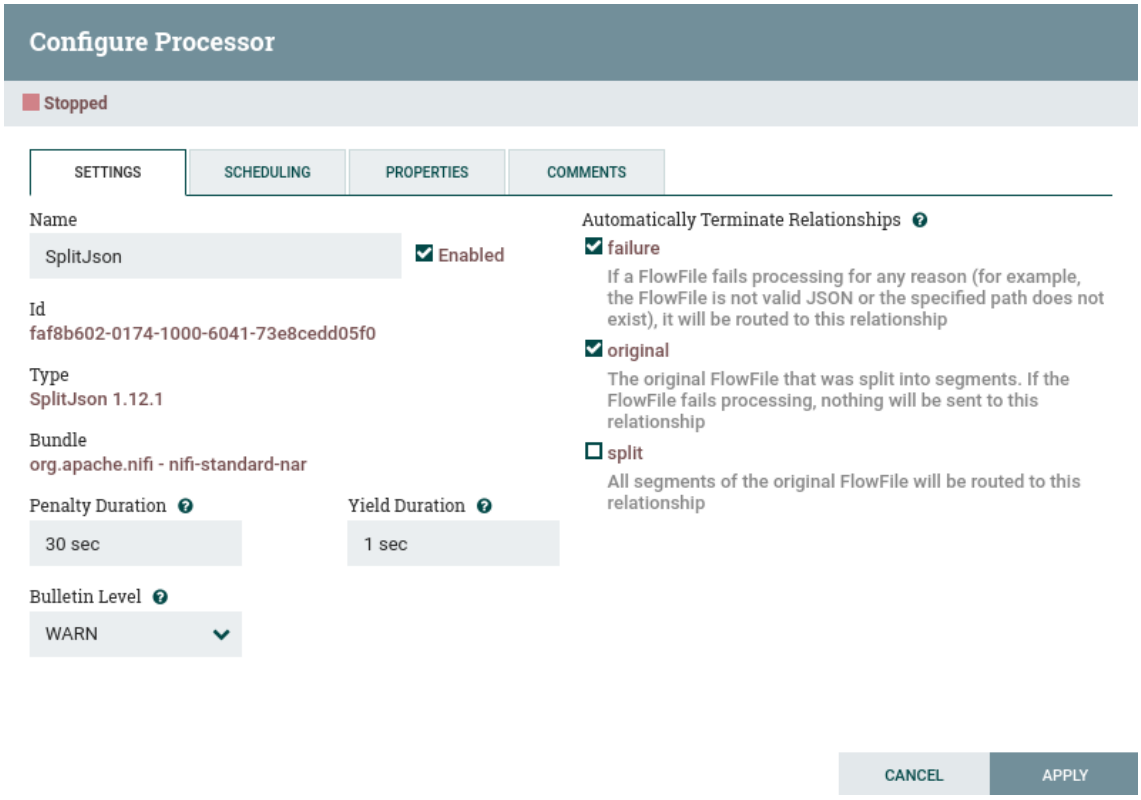
Arrastramos un Procesador UpdateAttribute para establecer el nombre del fichero al valor del atributo municipio_distrito con extensión *csv*. Esto se hace actualizando el atributo filename. Luego, creamos una Conexión entre MergeContent y UpdateAttribute para la Relación merged.



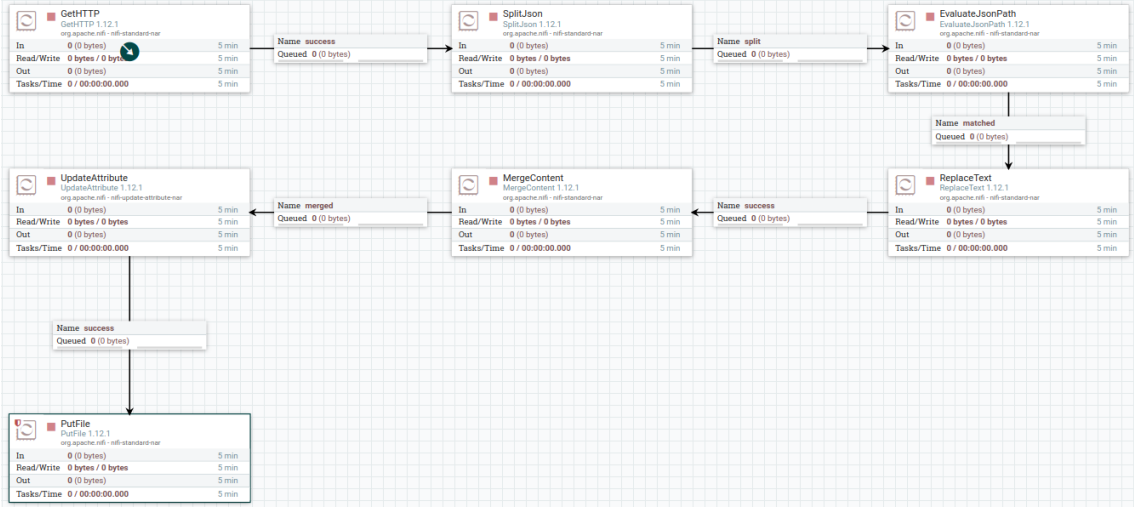
Añadimos un Procesador PutFile para guardar los CSVs en una carpeta local. Especificamos la propiedad Directory y cambiamos Conflict Resolution Strategy a replace para que reemplace el CSV si este ya existe. Luego, creamos una Conexión entre UpdateAttribute y PutFile para la relación success.



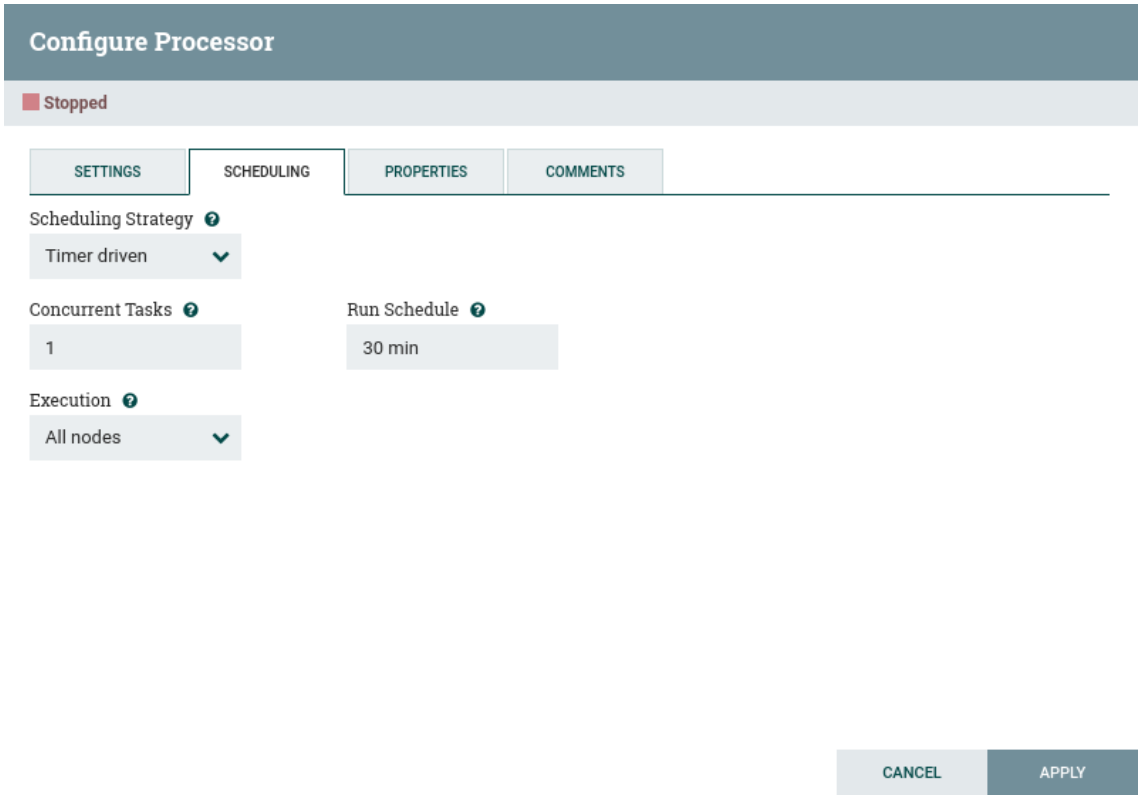
Por último, para cada Procesador tenemos que terminar todas las Relaciones que no se usan en Conexiones. Por ejemplo, para el Procesador SplitJson hay que terminar las relaciones failure y original.



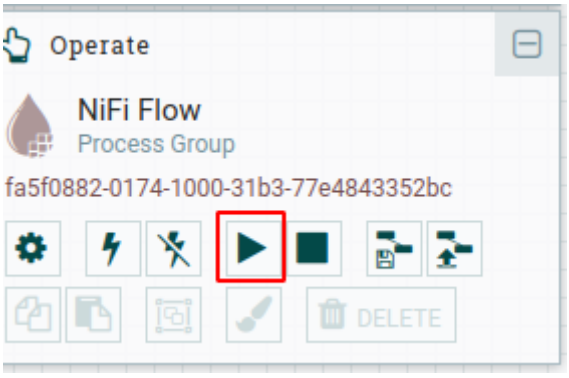
El flujo debe quedar así:



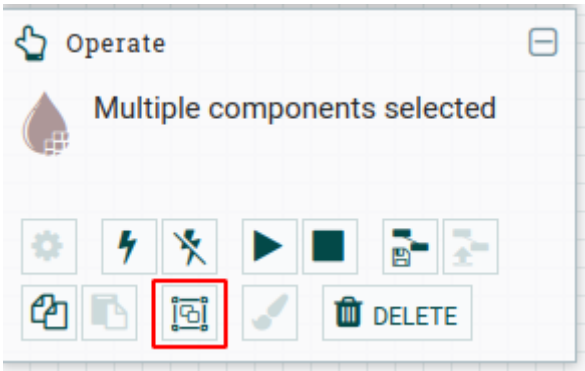
Los Procesadores pueden ser programados usando temporizadores o CRONes. Programamos GetHTTP para que se ejecute cada 30 minutos.



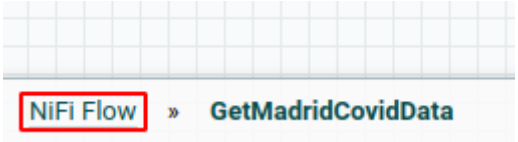
Podemos poner el flujo en marcha, bien haciendo click en el lienzo, o bien seleccionando los Procesadores que queremos ejecutar y luego el botón Start de la Paleta Operate.



Tiene sentido agrupar este flujo en un Grupo de Procesos (Process Group). Para agruparlos tenemos que seleccionar todo y hacer click en el botón Group de la paleta Operate.



Una vez hecho esto podemos hacer doble click en el Grupo de Procesos para ver y modificar los componentes. Podemos navegar por los Grupos de Procesos usando la barra de navegación que se encuentra en la parte inferior del lienzo. Por ejemplo, para navegar al dataflow principal hacemos click en *NiFi Flow*.



6. MINIFI

MiNiFi es un subproyecto de Apache NiFi enfocado a la recopilación de datos.

Los objetivos de MiNiFi son:

- Tamaño pequeño y bajo consumo de recursos.
- Gestión centralizada de agentes.
- Generación de procedencia de datos.
- Integración con NiFi para la gestión del flujo de datos.

MiNiFi tiene agentes de Java y C++. El agente de Java puede ejecutar la mayoría de los Procesadores disponibles de NiFi, pero es una distribución binaria más grande y consume más recursos del sistema. El agente C ++ es un binario más pequeño, consume poca memoria del sistema, pero tiene un subconjunto limitado de Procesadores.

MiNiFi Java requiere Java 1.8+ mientras que MiNiFi C++ requiere las siguientes librerías:

- RHEL/CentOS:

```
yum install -y epel-release
yum install -y leveldb
```

- Debian/Ubuntu:

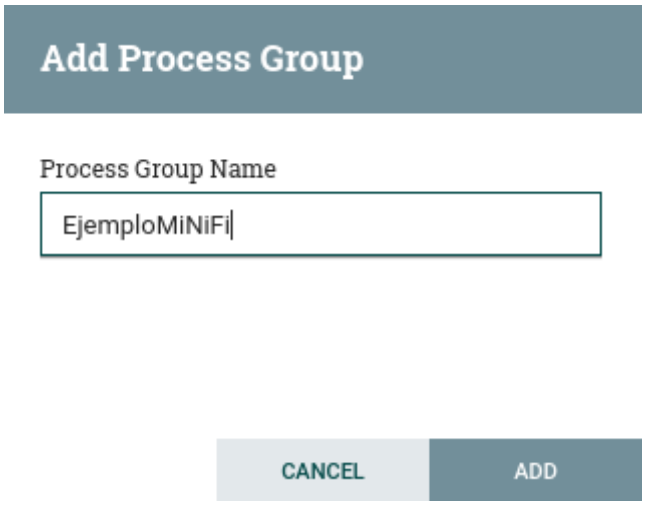
```
apt install -y libleveldb-dev
apt install -y libxml2
```

Los agentes se encuentran en la página de descargas de MiNiFi.

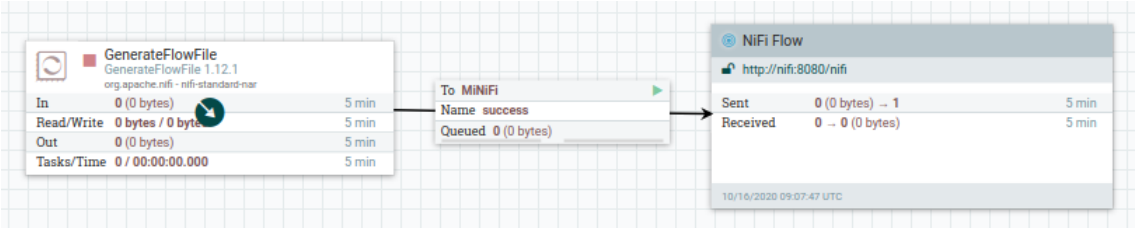
Una vez descomprimido el archivo hay que establecer la variable de entorno MINIFI_HOME.

Los agentes ejecutan el flujo definido en *$MINIFI_HOME/conf/config.yml*. Podemos crear el flujo en la interfaz web de Apache NiFi y luego guardarlo como una plantilla (Template), descargarlo y convertirlo a *config.yml* con Converter Toolkit.

Primero, creamos un Grupo de Procesos.

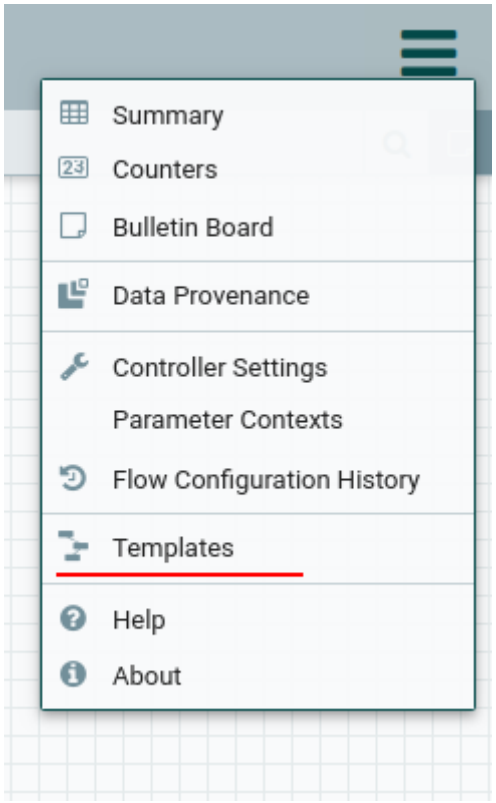


Añadimos un Procesador GenerateFlowFile y lo conectamos con un Grupo Remoto de Procesos.



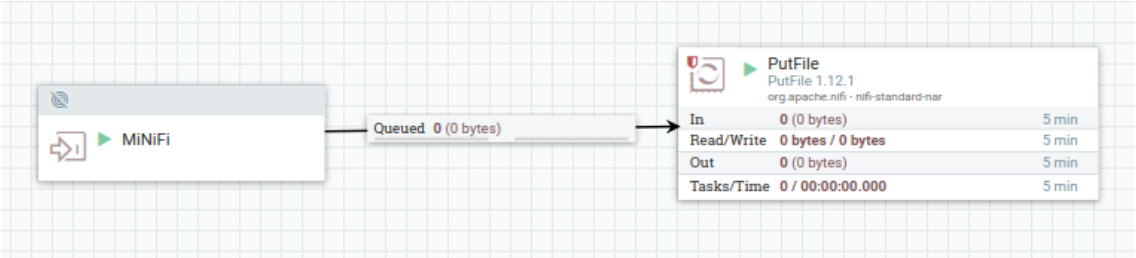
Guardamos este Grupo de Procesos como plantilla haciendo click en el lienzo y luego *Create template*.

Una vez creada, la descargamos yendo al *Menú Global Templates*.



En la parte derecha pinchamos en el botón de descarga.

Creamos otro flujo que va a recibir los datos de MiNiFi, arrastrando un Input Port y un Procesador PutFile, creando la conexión entre ellos.



Transformamos el XML de la plantilla en config.yml.

```
config.sh transform MiNiFi.xml config.yml
```

Movemos el nuevo fichero en *$MINIFI_HOME/conf/*.

Ejecutamos MiNiFi.

```
bin/minifi.sh run
```

7. CASOS DE USO

Apache NiFi se creó para hacer una cosa muy bien y esa es la gestión de flujos de datos. Como ya existen muchos sistemas que se enfocan al procesamiento de datos (Apache Spark, Flink, herramientas ETLs como Talend, PDI...), sistemas de almacenamiento de datos (bases de datos, HDFS...), orígenes de datos (dispositivos, sensores, ficheros locales, REST...), Apache NiFi se enfoca exclusivamente en la tarea de conectar esos sistemas y brindar la experiencia del usuario y las funciones básicas necesarias para hacerlo bien.

Las funciones clave de NiFi son:

- Control interactivo - a la hora de conectar sistemas es importante poder interactuar rápido y eficientemente con los flujos constantes de datos. La interfaz de usuario de NiFi nos permite hacer precisamente eso a medida que fluyen los datos, podemos agregar funciones para operar con ellos, bifurcar copias de datos para probar nuevos enfoques, ajustar la configuración actual, ver estadísticas recientes e históricas... En comparación, casi todos los demás sistemas tienen un modelo que está orientado al diseño y la implementación. Ese modelo está bien y puede ser intuitivo, pero para la administración de flujo de datos significa que no obtenemos la retroalimentación cambio por cambio que es tan vital para construir rápidamente nuevos flujos o para corregir o mejorar de manera segura y eficiente el manejo de los flujos de datos existentes.
- Procedencia de datos - NiFi proporciona la capacidad para generar detalles de trazabilidad detallados y potentes de dónde provienen sus datos, qué se les hace, dónde se envían y cuándo se realizan en el flujo. Esto es esencial para una gestión eficaz del flujo de datos por varias razones, pero para alguien que se encuentra en las primeras fases de exploración y trabaja en un proyecto, lo más importante que le brinda es una increíble flexibilidad de depuración. Puede configurar sus flujos y dejar que las cosas funcionen y luego usar la procedencia para demostrar realmente que hizo exactamente lo que quería. Si algo no sucedió como esperaba, puede arreglar el flujo y reproducir el objeto y luego repetir.
- Schemaless - NiFi no fuerce la conversión de datos a algún formato especial de NiFi ni fuerce la reconversión a algún formato para la entrega posterior. Esto significa que es fácil de manejar diversos conjuntos de datos (imágenes, videos, audio y más). Con NiFi podemos manejar datos estructurados, semiestructurados y no estructurados con la misma facilidad.

Podemos decir que Apache NiFi es un acelerador de proyectos de Big Data donde el objetivo es ingerir una variedad de fuente de datos. Tener Apache NiFi como una única plataforma de ingesta que le brinda herramientas listas para usar para ingerir varias fuentes de datos de manera segura y controlada acelera la disponibilidad de datos en un Data Lake o Data Warehouse y, por tanto, acelera el proyecto de Big Data y la extracción de valor comercial.

Se puede usar para:

- Ingesta y enrutamiento de datos IoT en tiempo real.
- Validación de datos en tiempo real.
- Mover datos entre varios tipos de sistemas.
- Ingesta y enriquecimiento de datos.

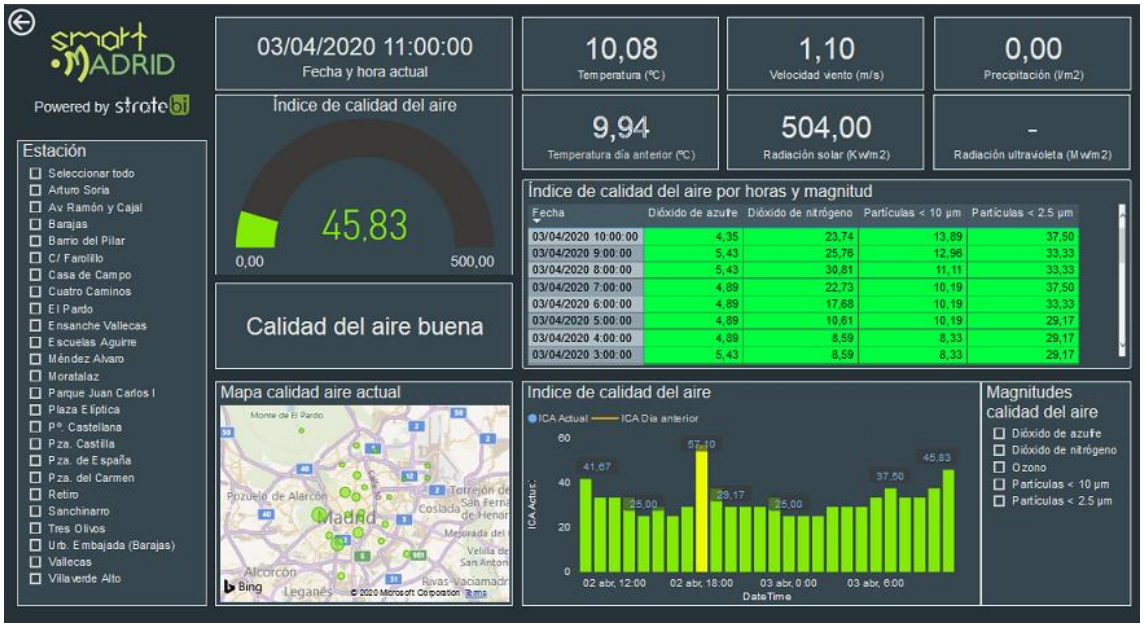
8. CONCLUSIÓN

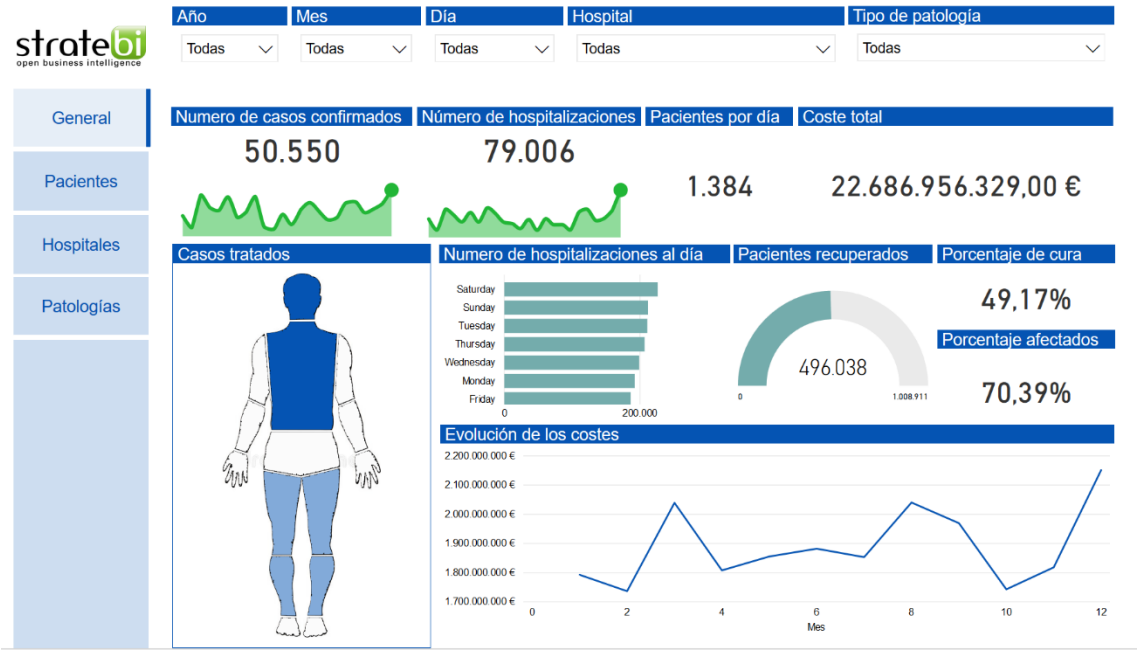
Apache NiFi es un sistema para gestionar flujos de datos complejos. Su programación basada en flujo hace que sea fácil de usar y aprender. Es una aplicación multiplataforma fácil de instalar, configurar y es altamente escalable y flexible.

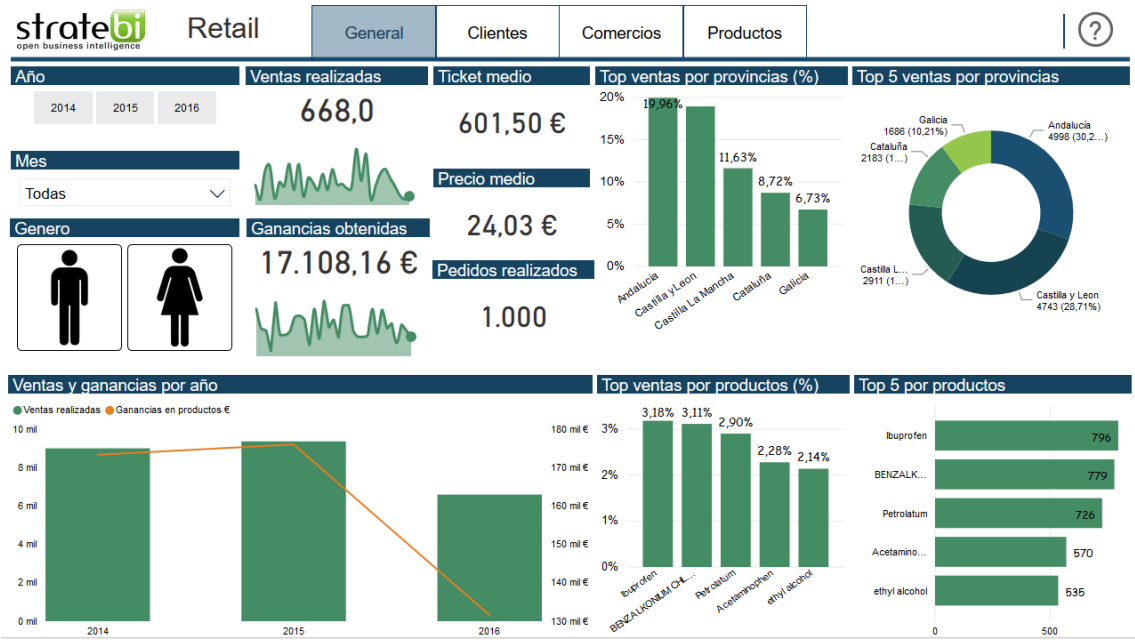
Aunque se puede usar para ETLs ligeros, su mayor uso es para la ingesta, el enriquecimiento y enrutamiento de datos.

9. POWER BI

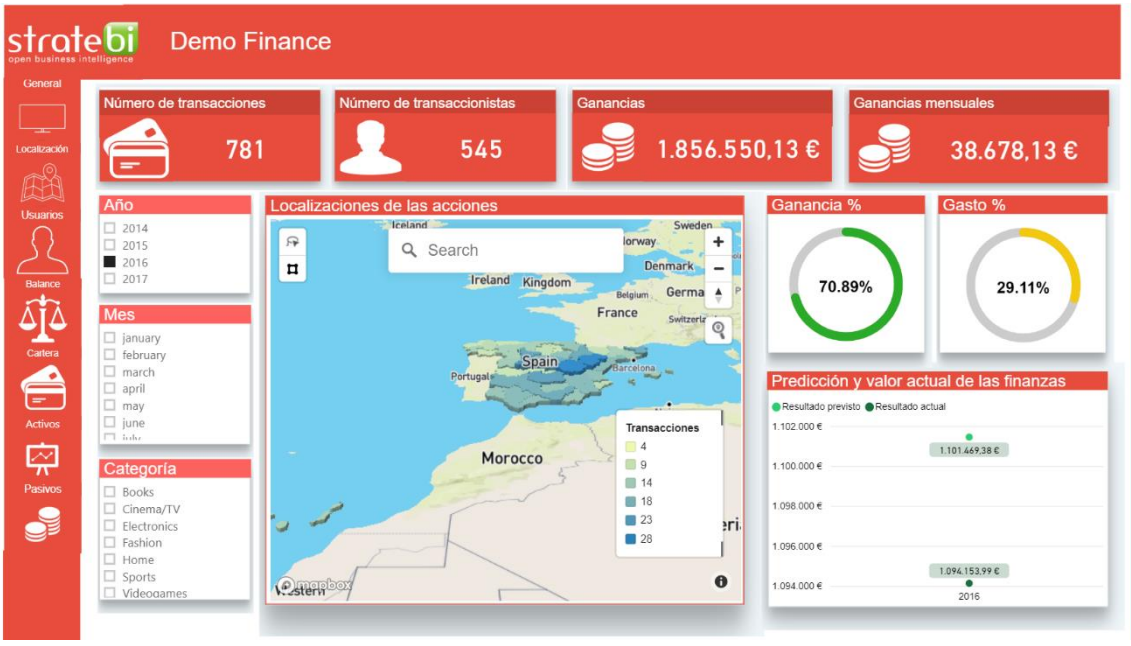
Stratebi es también Partner Certificado en Microsoft Power BI. En esta sección puedes consultar algunas Demos Online en donde ver el potencial de la herramienta, así como algunos videotutoriales

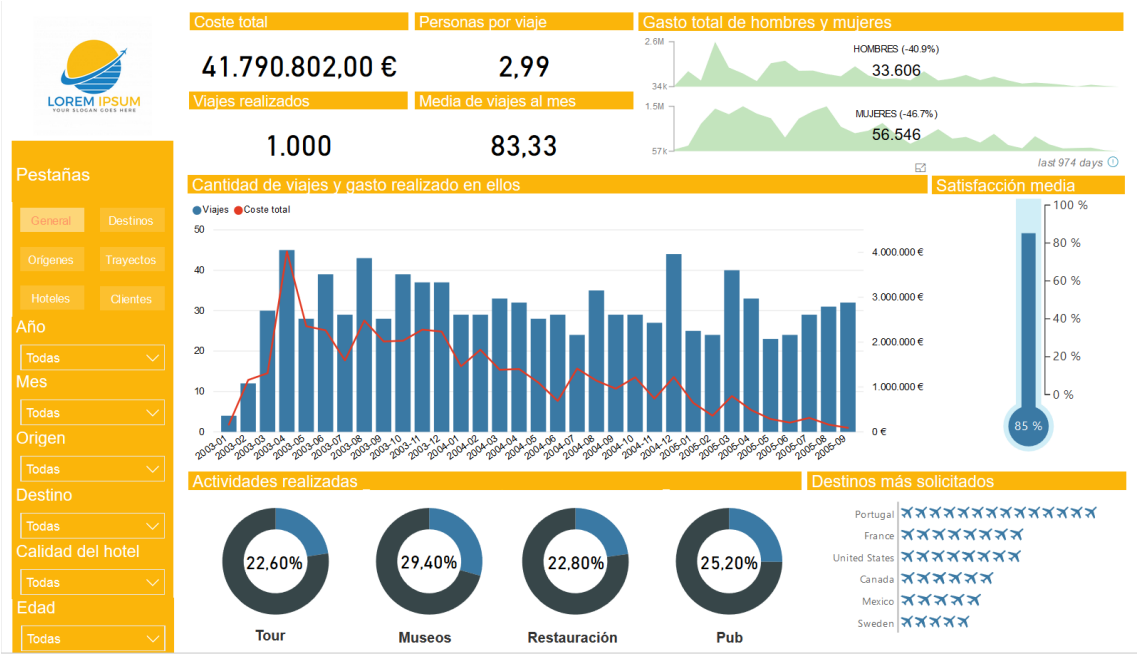



Año
Mes
Día
Hospital
Tipo de patología
Todas
General
Pacientes
Hospitales
Patologías
Numero de casos confirmados
50.550
Número de hospitalizaciones
79.006
Pacientes por día
1.384
Coste total
22.686.956.329,00 €
Casos tratados
Numero de hospitalizaciones al día
Saturday
Sunday
Tuesday
Thursday
Wednesday
Monday
Friday
Pacientes recuperados
496.038
Porcentaje de cura
49,17%
Porcentaje afectados
70,39%
Evolución de los costes
Mes

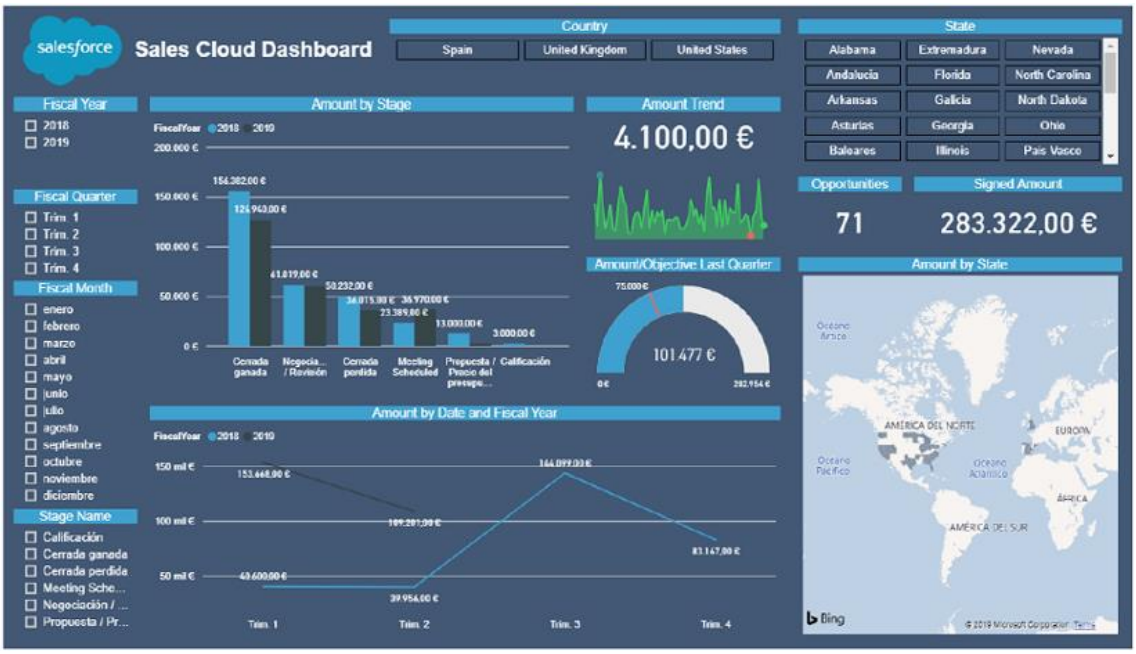

Retail
General
Clientes
Comercios
Productos
Año
2014
2015
2016
Mes
Todas
Genero
Ventas realizadas
668,0
Ganancias obtenidas
17.108,16 €
Ticket medio
601,50 €
Precio medio
24,03 €
Pedidos realizados
1.000
Top ventas por provincias (%)
19,96%
11,63%
8,72%
6,73%
Andalucia
Castilla y Leon
Castilla La Mancha
Cataluña
Galicia
Top 5 ventas por provincias
Galicia 1686 (10,21%)
Andalucia 4998 (30,2...)
Cataluña 2183 (1...)
Castilla L... 2911 (1...)
Castilla y Leon 4743 (28,71%)
Ventas y ganancias por año
Ventas realizadas
Ganancias en productos €
2014
2015
2016
Top ventas por productos (%)
3,18%
3,11%
2,90%
2,28%
2,14%
Ibuprofen
BENZALKONIUM CHL...
Petrolatum
Acetaminophen
ethyl alcohol
Top 5 por productos
Ibuprofen 796
BENZALK... 779
Petrolatum 726
Acetamino... 570
ethyl alcohol 535


Year
2019
Financial Results - Graphs
stratebi open business intelligence
Earnings
Direct Expenses
Gross Margin
Commercial Expenses
Contribution Margin
EBITDA
Corporative EBITDA
EBIT

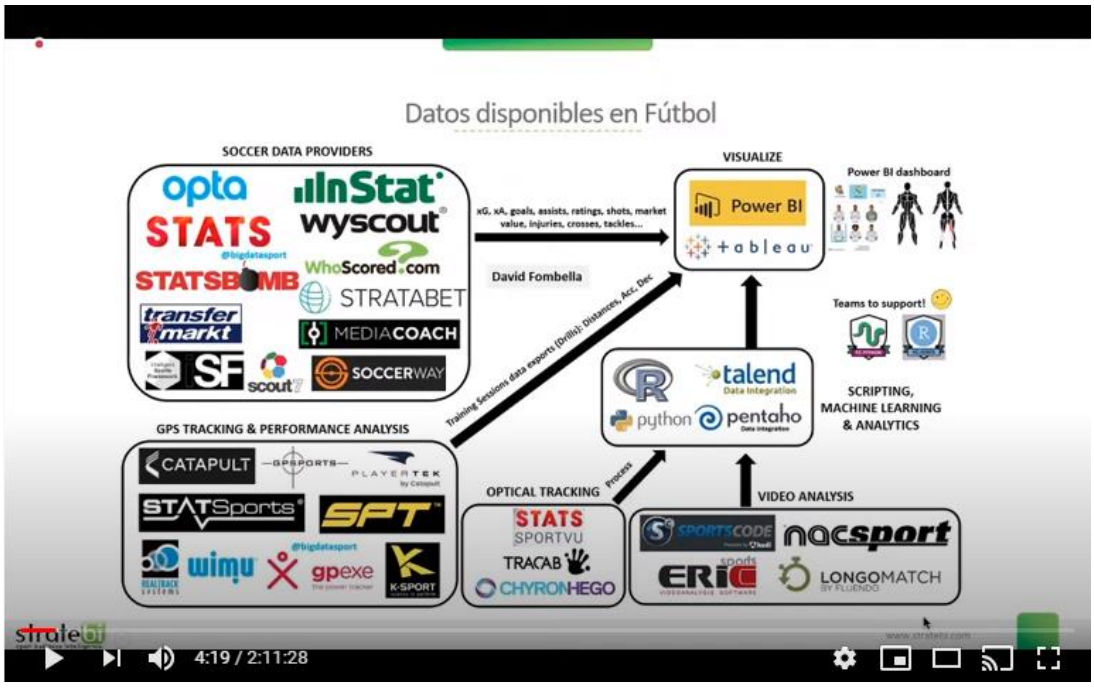

stratebi open business intelligence
Demo Finance
General
Localización
Usuarios
Balance
Cartera
Activos
Pasivos
Número de transacciones
781
Número de transaccionistas
545
Ganancias
1.856.550,13 €
Ganancias mensuales
38.678,13 €
Año
2014
2015
2016
2017
Mes
january
february
march
april
may
june
Categoría
Books
Cinema/TV
Electronics
Fashion
Home
Sports
Videogames
Localizaciones de las acciones
Search
Transacciones
4
9
14
18
23
28
Ganancia %
70.89%
Gasto %
29.11%
Predicción y valor actual de las finanzas
Resultado previsto
Resultado actual
1.101.469,38 €
1.094.153,99 €
2016


LOREM IPSUM
Pestañas
General
Destinos
Orígenes
Trayectos
Hoteles
Clientes
Año
Todas
Mes
Todas
Origen
Todas
Destino
Todas
Calidad del hotel
Todas
Edad
Todas
Coste total
41.790.802,00 €
Personas por viaje
2,99
Viajes realizados
1.000
Media de viajes al mes
83,33
Gasto total de hombres y mujeres
HOMBRES (-40.9%)
33.606
MUJERES (-46.7%)
56.546
last 974 days
Cantidad de viajes y gasto realizado en ellos
Viajes
Coste total
Satisfacción media
85 %
Actividades realizadas
22,60%
Tour
29,40%
Museos
22,80%
Restauración
25,20%
Pub
Destinos más solicitados
Portugal
France
United States
Canada
Mexico
Sweden


PowerLinceBI
FÚTBOL
ANALYTICS
Top 10 jugadores
Mapa Jugadores
Mapa Formaciones y Demarcación
Formaciones Equipo
Estadísticas Equipo
Estadísticas Jugador
Ayuda
Creado por LinceBI


salesforce
Sales Cloud Dashboard
Country
Spain
United Kingdom
United States
State
Alabama
Extremadura
Nevada
Andalucía
Florida
North Carolina
Arkansas
Galicia
North Dakota
Asturias
Georgia
Ohio
Baleares
Illinois
País Vasco
Fiscal Year
2018
2019
Fiscal Quarter
Trim. 1
Trim. 2
Trim. 3
Trim. 4
Fiscal Month
enero
febrero
marzo
abril
mayo
junio
julio
agosto
septiembre
octubre
noviembre
diciembre
Stage Name
Calificación
Cerrada ganada
Cerrada perdida
Meeting Sche...
Negociación / ...
Propuesta / Pr...
Amount by Stage
Amount Trend
4.100,00 €
Opportunities
71
Signed Amount
283.322,00 €
Amount/Objective Last Quarter
101.477 €
Amount by State
Amount by Date and Fiscal Year
Trim. 1
Trim. 2
Trim. 3
Trim. 4


Tutorial de Introducción a Power BI
Ver más tarde
Compartir



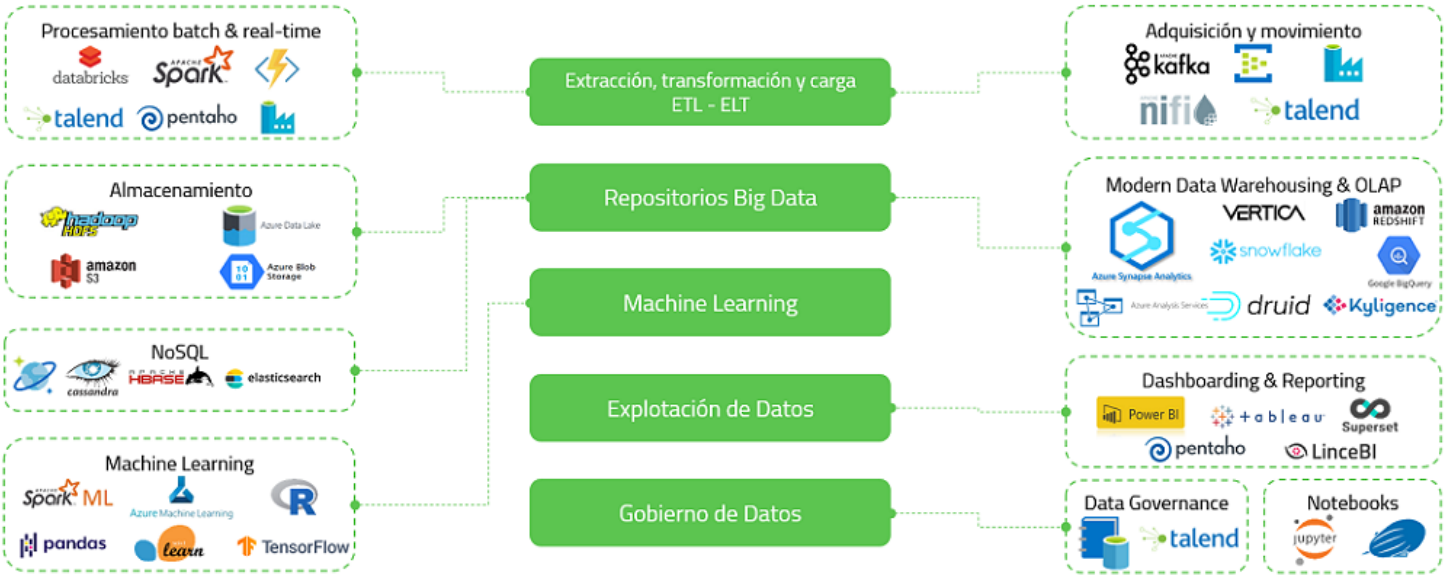
Sports Analytics con PowerBI

Más Videotutoriales y Manuales sobre Power BI:

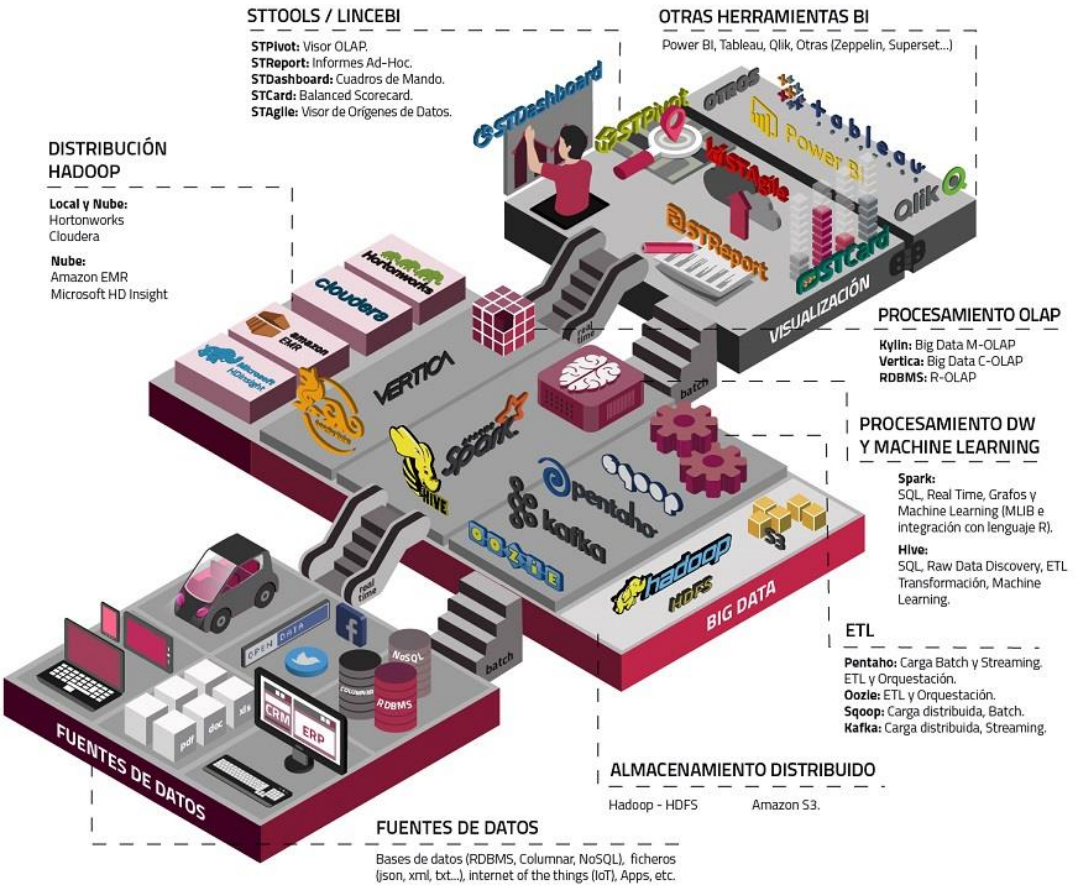
1. ALM Toolkit para PowerBI
2. Tabular Editor con PowerBI
3. PowerBI Embeded – Buenas Prácticas
4. PowerBI Tips (vol I)
5. PowerBI Tips (vol II)
6. Buenas prácticas con Dataflows en PowerBI
7. Integracion SAP - PowerBI
8. Futbol Analytics, lo que hay que saber
9. Dashboard de medicion de la calidad del aire en Madrid
10. Como funciona Microsoft Power BI? Videoturial de Introducción
11. Big Data para PowerBI
12. Como integrar Salesforce y PowerBI
13. Videotutorial: Usando R para Machine Learning con PowerBI
14. Las 50 claves para aprender y conocer PowerBI
15. PowerBI: Arquitectura End to End
16. Usando Python con PowerBI
17. PowerBI + Open Source = Sports Analytics
18. Comparativa de herramientas Business Intelligence
19. Use Case Big Data "Dashboards with Hadoop and Power BI"
20. Todas las presentaciones del Workshop 'El Business Intelligence del Futuro'
21. Descarga Paper gratuito: Zero to beautiful (Data visualization)
22. DAX Editor para Power BI

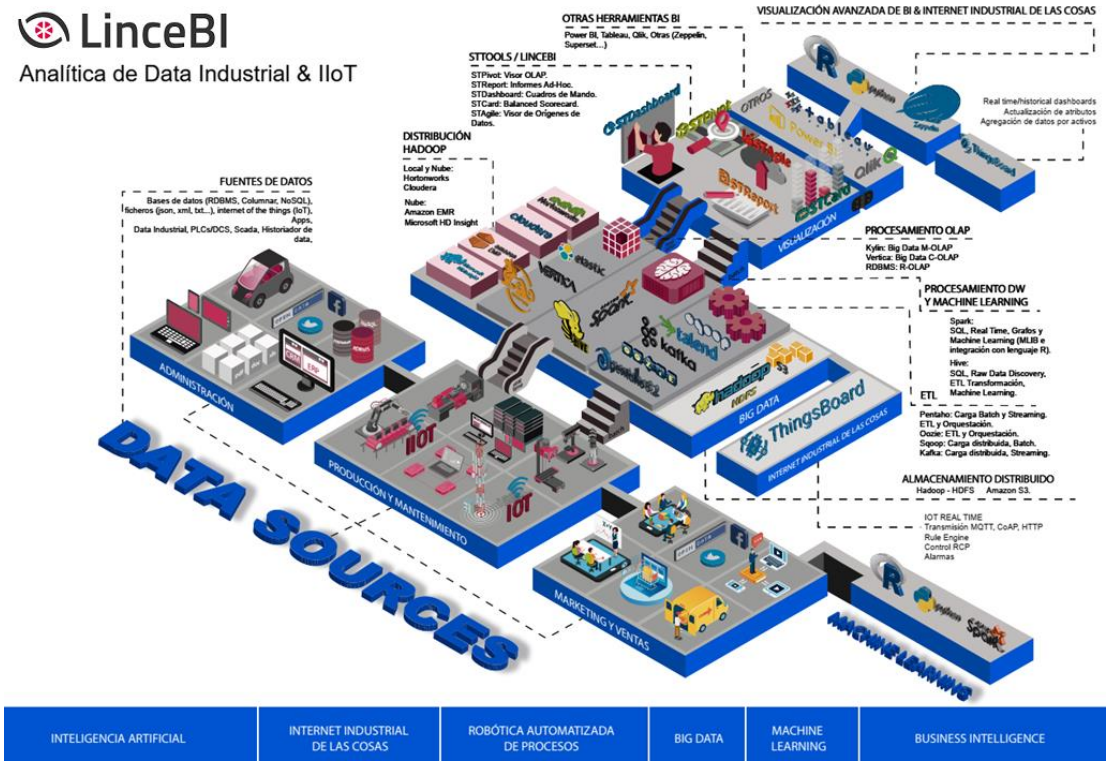
10. TECNOLOGÍAS

Recientemente, hemos sido nombrados Partners Certificados de Vertica, Talend, Microsoft, Snowflake, Kylligence, Pentaho, etc.



Aplicaciones y tecnologías en las que somos especialistas


STTOOLS / LINCEBI
STPivot: Visor OLAP.
STReport: Informes Ad-Hoc.
STDashboard: Cuadros de Mando.
STCard: Balanced Scorecard.
STAgile: Visor de Orígenes de Datos.
OTRAS HERRAMIENTAS BI
Power BI, Tableau, Qlik, Otras (Zeppelin, Superset...)
DISTRIBUCIÓN HADOOP
Local y Nube:
Hortonworks
Cloudera
Nube:
Amazon EMR
Microsoft HD Insight
VISUALIZACIÓN
PROCESAMIENTO OLAP
Kylin: Big Data M-OLAP
Vertica: Big Data C-OLAP
RDBMS: R-OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
Spark:
SQL, Real Time, Grafos y Machine Learning (MLIB e integración con lenguaje R).
Hive:
SQL, Raw Data Discovery, ETL Transformación, Machine Learning.
BIG DATA
ETL
Pentaho: Carga Batch y Streaming. ETL y Orquestación.
Oozie: ETL y Orquestación.
Sqoop: Carga distribuida, Batch.
Kafka: Carga distribuida, Streaming.
FUENTES DE DATOS
ALMACENAMIENTO DISTRIBUIDO
Hadoop - HDFS
Amazon S3.
FUENTES DE DATOS
Bases de datos (RDBMS, Columnar, NoSQL), ficheros (json, xml, txt...), internet of the things (IoT), Apps, etc.


LinceBI
Analítica de Data Industrial & IIoT
OTRAS HERRAMIENTAS BI
Power BI, Tableau, Qlik, Otras (Zeppelin, Superset...)
VISUALIZACIÓN AVANZADA DE BI & INTERNET INDUSTRIAL DE LAS COSAS
STTOOLS / LINCEBI
STPivot: Visor OLAP.
STReport: Informes Ad-Hoc.
STDashboard: Cuadros de Mando.
STCard: Balanced Scorecard.
STAgile: Visor de Orígenes de Datos.
DISTRIBUCIÓN HADOOP
Local y Nube:
Hortonworks
Cloudera
Nube:
Amazon EMR
Microsoft HD Insight
FUENTES DE DATOS
Bases de datos (RDBMS, Columnar, NoSQL), ficheros (json, xml, txt...), internet of the things (IoT), Apps.
Data Industrial, PLCs/DCS, Scada, Historiador de data.
Real time/historical dashboards
Actualización de atributos
Agregación de datos por activos
PROCESAMIENTO OLAP
Kylin: Big Data M-OLAP
Vertica: Big Data C-OLAP
RDBMS: R-OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
Spark:
SQL, Real Time, Grafos y Machine Learning (MLIB e integración con lenguaje R).
Hive:
SQL, Raw Data Discovery, ETL Transformación, Machine Learning.
ETL
Pentaho: Carga Batch y Streaming. ETL y Orquestación.
Oozie: ETL y Orquestación.
Sqoop: Carga distribuida, Batch.
Kafka: Carga distribuida, Streaming.
ALMACENAMIENTO DISTRIBUIDO
Hadoop - HDFS
Amazon S3.
IOT REAL TIME
Transmisión MQTT, CoAP, HTTP
Rule Engine
Control RCP
Alarmas
ADMINISTRACIÓN
PRODUCCIÓN Y MANTENIMIENTO
MARKETING Y VENTAS
MACHINE LEARNING
DATA SOURCES
INTELIGENCIA ARTIFICIAL
INTERNET INDUSTRIAL DE LAS COSAS
ROBÓTICA AUTOMATIZADA DE PROCESOS
BIG DATA
MACHINE LEARNING
BUSINESS INTELLIGENCE

11. INFORMACIÓN SOBRE STRATEBI

Stratebi es una empresa española, con sede en Madrid y oficinas en Barcelona, Alicante y Sevilla, con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes.

Somos **Partners Certificados en Microsoft PowerBI** con una dilatada experiencia

Stratebi es la única empresa española que ha estado presente todos los Pentaho Developers celebrados en Europa habiendo organizado el de España.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi **son profesores y responsables de proyectos del Master en Business Intelligence de la Universidad UOC, UCAM, EOI…**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source. Todobi.com

Stratebi es partner de las principales soluciones Analytics: Microsoft Power BI, Talend, Pentaho, Vertica, Snowflake, Kyligence, Cloudera…

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

12. OTROS

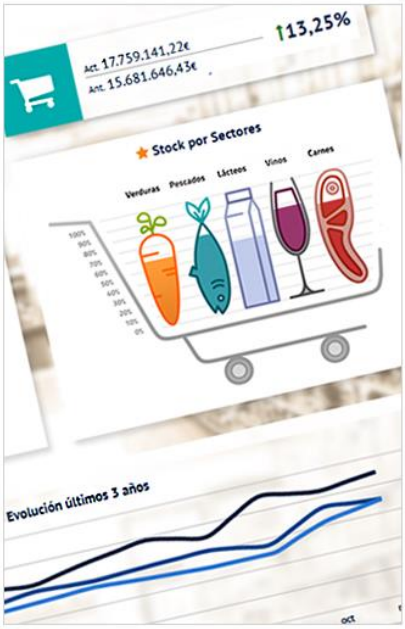
Trabajamos en los principales sectores y con algunas de las compañías y organizaciones más importantes de España.

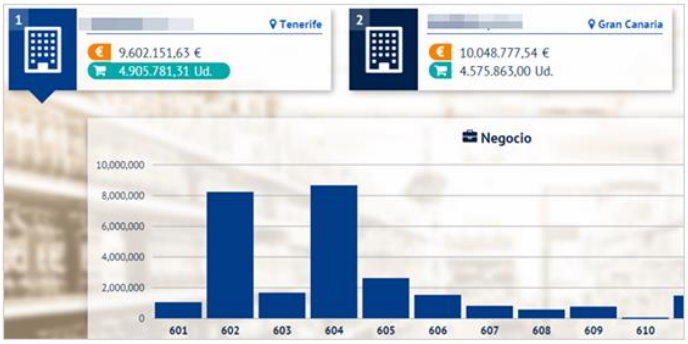
SECTOR PRIVADO

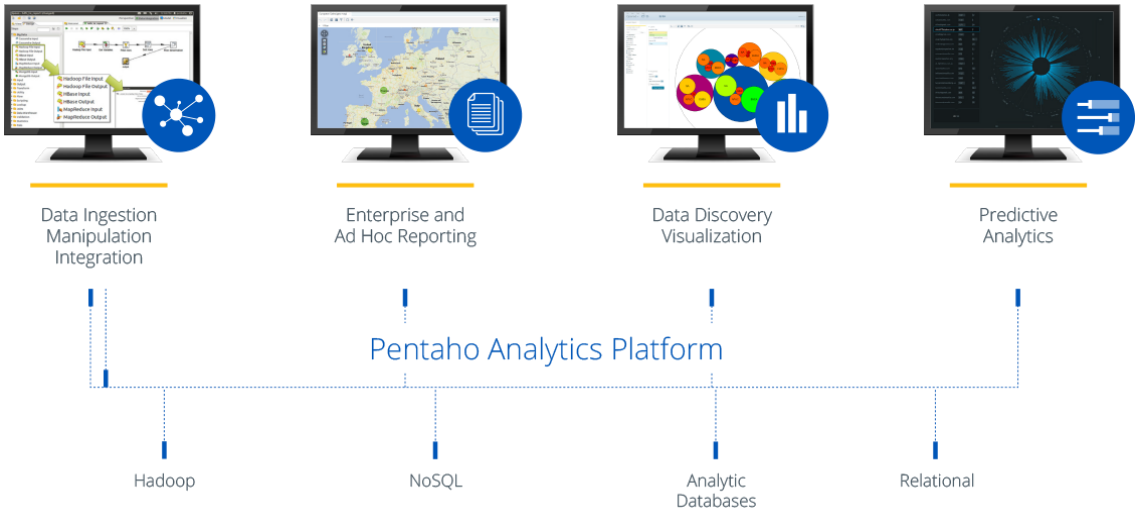
SECTOR PÚBLICO

13. EJEMPLOS DE DESARROLLOS ANALYTICS

A continuación, se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:

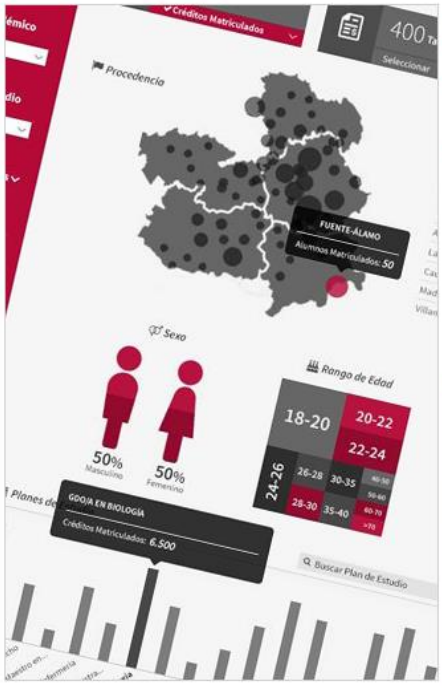




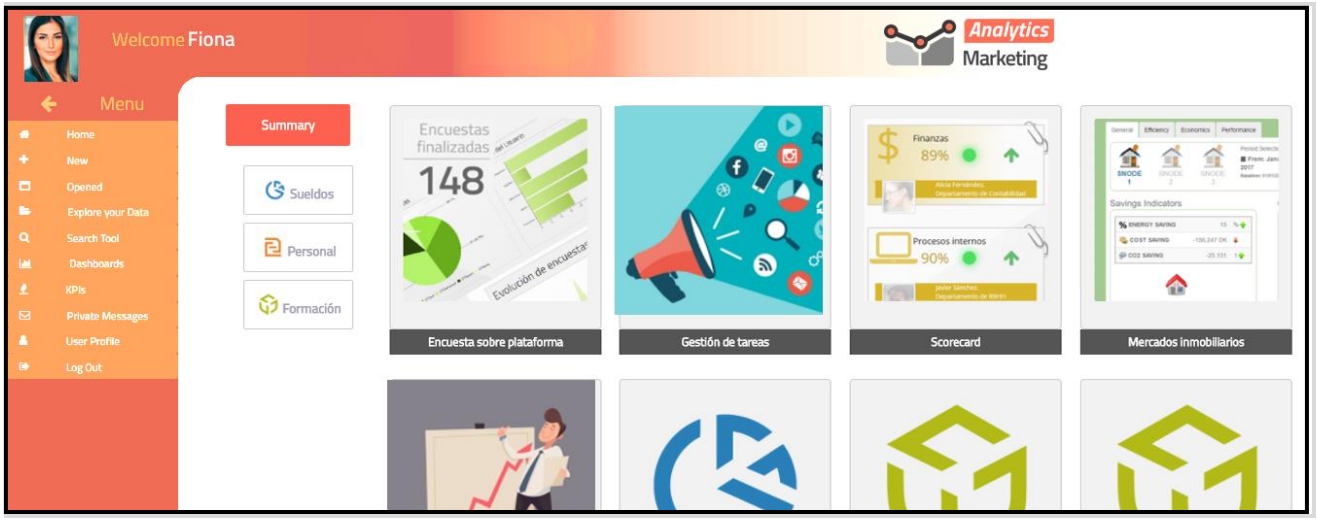



facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
65%
35%
28%
60%
12%
Tipo de recurso
Evolución Comentarios por días


smart MADRID
Último Actualización
18:24
TOP 5 PUNTOS DE MEDIDA
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50%
50%
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GRADO EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio


181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU
JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Evolución de encuestas
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Savings Indicators
Mercados inmobiliarios