Análisis de datos previo a la implantación de las zonas de bajas emisiones (ZBE) en diferentes municipios de la provincia de Barcelona

Elaboración de lago de datos para analizar la relación contaminación-climatología basado en el análisis de datos abiertos disponibles por la Generalidad de Catalunya

Oscar Hidalgo Fernandez

Programas de Inteligencia de Negocio, Big Data, Análisis y Estrategia de Datos - Área Big Data

Tutor de TF: **Juan Carlos Castro Robles**
Profesores responsables de la asignatura: **Josep Curto y Atanasi Daradoumis**
Fecha de entrega: **06/2024**

Ficha de trabajo final de máster:

Título del trabajo:	Análisis de datos previo a la implantación de las zonas de bajas emisiones (ZBE) en diferentes municipios de la provincia de Barcelona
Nombre del autor:	Oscar Hidalgo Fernandez
Nombre del Tutor del TF:	Juan Carlos Castro Robles
Nombre del PRA:	Josep Curto y Atanasi Daradoumis
Fecha de entrega:	06/2024
Titulación o programa:	Programas de Inteligencia de Negocio, Big Data, Análisis y Estrategia de Datos - Big Data
Área del Trabajo Final:	Big Data - Data Analytics - Business Intelligence
Idioma del trabajo:	Castellano
Palabras clave:	Lago de datos, Polución, Climatología

Resumen del Trabajo

Análisis de la evolución de la contaminación en diferentes municipios de la provincia de Barcelona. Basándose inicialmente en los contaminantes CO, NO, PM2.5 y PM10. Posibilitando la ampliación del número de contaminantes en el sistema de cara a futuro. Aportando el enfoque de la relación con variables climatológicas y temporales.

Se realiza para preparar a los diferentes municipios ante la instauración de las zonas de bajas emisiones de contaminación y que se puedan tomar las medidas oportunas.

Para ello se ha elaborado un lago de datos de extremo a extremo. De forma automatizada capta los datos de fuentes Open Data (sensores de contaminación y climatológicos). Se almacena en un entorno preparado para Big Data. Los procesa preparando con antelación posibles consultas a realizar y por último se realiza una presentación en forma de cuadro de mandos para que el usuario pueda obtener respuestas. Utilizando una ventana temporal que abarca hasta dos años previos.

Se ha observado las diferentes evoluciones tanto ascendentes (en unas) como descendentes (en otras) de la contaminación en las ciudades escogidas para la prueba de concepto y como la contaminación está asociada a las variables climatológicas escogidas.

Ha sido posible constatar el hecho de que es posible utilizar los conocimientos adquiridos durante el desarrollo del máster para el manejo de un gran volumen de datos en casi tiempo real para obtener respuestas a necesidades planteadas por los usuarios.

Abstract

Analysis of the evolution of pollution in different municipalities in the province of Barcelona. Initially based on the pollutants CO, NO, PM2.5 and PM10. Enabling the expansion of the number of pollutants in the system in the future. Providing the approach of the relationship with climatological and temporal variables.

It is carried out to prepare the different municipalities for the establishment of low pollution emission zones so that the appropriate measures can be taken.

For this, an end-to-end data lake has been created. It automatically captures data from OpenData sources (pollution and weather sensors). Stores them in a Big Data-ready environment. It processes them by preparing possible queries to be asked in advance and finally a presentation is made in the form of a dashboard so that the user can obtain answers. Using a time window that covers up to two previous years.

The different evolutions have been observed, both upward (in some) and downward (in others) of pollution in the cities chosen for the proof of concept and how pollution is associated with the chosen climatological variables.

It has been possible to verify the fact that it is possible to use the knowledge acquired during the development of the master's degree to manage a large volume of data in almost real time to obtain answers to needs raised by users.

Índice

1-Introducción trabajo

1.1-Contexto y justificación del trabajo

Actualmente, ya existen varias ZBE en Cataluña, especialmente en el ámbito metropolitano. La primera ZBE implementada fue la ZBE Rondes Barcelona y posteriormente se fue extendiendo a municipios del entorno cercano. Muchos ayuntamientos de grandes y medianas ciudades también están estudiando implantar ZBE para mejorar la calidad del aire y la salud de nuestras ciudades y para dar respuesta al marco normativo vigente.

El marco normativo vigente establece que las ciudades de más de 50.000 habitantes y las de más de 20.000 habitantes con superaciones de los valores límite de calidad del aire deben adoptar planes de movilidad urbana sostenible que introduzcan medidas de mitigación para reducir las emisiones derivadas de la movilidad incluyendo el establecimiento de ZBE antes del año 2023.

El BOE (#1.1) marcó como prioridad la política ambiental marcando límites en la contaminación para preservar la salud humana.

De cara a poder aplicar ZBE o poder tomar decisiones basadas en el dato para el control de la polución se necesita una solución que permita evaluar los datos disponibles para facilitar la tarea al usuario y obtener las respuestas necesarias. Adicionalmente también se necesita saber si las decisiones tomadas son las adecuadas permitiendo un seguimiento de la evolución de la polución.

De forma simultánea también se hace necesario saber la influencia del clima sobre la contaminación para poder predecir la evolución ante cambios de condiciones climatológicas.

1.1.1-Contexto según la OMS

Según la OMS (#1.2) actualmente el 99% de la población vive en áreas donde la contaminación supera las directrices marcadas.
Siendo esto la causa de 4,2 millones de muertes anuales de forma directa.
Siendo el origen de esta contaminación fuentes no controlables (incendios forestales, catástrofes naturales) y otras controlables (vehículos, industria, combustión de materiales…). Habiendo mayor concentración en países de ingresos medios y bajos.
Como medida de referencia la OMS ofrece una descripción de los contaminantes y sus límites establecidos (más restrictivos que los indicados en el BOE)

1.1.2-Referencias bibliográficas investigadas

Con tal de iniciar el desarrollo de este proyecto hemos investigado diferentes trabajos y análisis realizados en la línea de este proyecto, para inspirarnos e intentar no reinventar la rueda.
Hemos analizado soluciones propuestas para el estudio de la calidad del aire por parte de estudios (#1.3), (#1.4), (#1.5), (#1.6), (#1.7), (#1.8), (#1.9)
También hemos revisado plataformas gubernamentales o profesionales como Miteco (#1.10) o Calíope (#1.11)

Tras revisar las fuentes, observamos que la mayoría están enfocados al análisis puro, alejado de mantenerlo en el tiempo. Con tal de dar respuesta a situaciones pasadas.

1.2-Objetivos del trabajo

El objetivo de este trabajo es la realización de una solución técnica para la necesidad que se nos plantea.

En lugar de un trabajo de análisis puro vamos a enfocar el proyecto como un proyecto final empresarial, de cariz técnico.

Para permitir el análisis del gran volumen de datos en el mínimo tiempo posible (acercándonos al tiempo real) no solamente vamos a utilizar/desarrollar una herramienta. Vamos a implementar una suite de varias herramientas que serán utilizadas con tal de conseguir este fin. Nos marcamos como objetivos los siguientes:

- Altamente configurable (se debe poder configurar fácilmente).
- Escalable.
- Facilidad para incluir nuevos orígenes de datos.
- Elementos fácilmente reemplazables. Ante un requisito técnico específico pueda ser sustituido un componente por otro.
- Reducción de costes al mínimo necesario. Entendiendo que la sostenibilidad no únicamente es ambiental, sino también económica.

Como objetivo adicional, pensamos utilizar esta memoria como guía de instalación y uso, definiendo al detalle los pasos realizados, para que puedan ser reproducidos los resultados en sistemas ajenos, no limitando el análisis al contenido en este documento.

1.3-Impacto en sostenibilidad

Uno de los motivos de escoger el tema de este TFM es precisamente su impacto en la sostenibilidad ambiental. El análisis de la contaminación es un tema preocupante y que debería estar en primer plano.

Una de las consecuencias históricas del progreso en la civilización es el aumento en la contaminación ambiental. Llegando a límites inaceptables para la salud humana en la época moderna. No haciéndose suficiente para paliarlo, basándose en razonamientos parciales, sofismas y falacias lógicas para justificar la inacción ante la situación.

Es solo mediante el dato claro, lógico y contrastado que se pueden obtener resultados que nos permitan tomar decisiones eficientes.

Facilitando el análisis de la evolución de la contaminación al usuario. Permitiendo ir desde el volumen más general al detalle más pequeño. Facilitando el establecer la relación con las condiciones meteorológicas adicionalmente, para saber que realmente influencia del clima en la contaminación.

Gracias a los resultados de este análisis se permite saber qué medidas tomadas en contra de la polución son efectivas y cuáles no de forma fidedigna.

1.4-Compromiso con privacidad del dato

En aras de seguir la legislación vigente, en relación la protección de los datos personales y reforzar la privacidad del mismo dato, nos hemos enfocado en utilizar datos totalmente anonimizados y alejados de cualquier dato que pueda identificar a una persona. Descartando fuentes de datos que pudieran facilitar identificaciones, como por ejemplo, los provenientes de cámaras de tráfico.

Nos centraremos en datos de origen gratuito y provenientes de fuentes totalmente abiertas. Siendo, en este caso, datos provenientes de la administración pública a partir de sus portales de transparencia.

1.5-Enfoque y método seguido

1.5.1-Enfoque técnico

Para realizar este proyecto se ha determinado que la mejor solución es la creación de un lago de datos. Un sistema de datos almacenados, también considerado un repositorio centralizado que engloba los diferentes niveles necesarios para su gestión (ingesta del dato, procesamiento y visualización/exportación del dato).

Se puede utilizar para datos estructurados, no estructurados o semiestructurados. Perfecto partiendo del desconocimiento del dato que vamos a encontrar para trabajar. Se despreocupa del tamaño ocupado y se preocupa de dejar el dato procesado para reducir al máximo el tiempo invertido en la consulta.

A lo largo de esta memoria vamos a desgranar las diferentes áreas que compondrán este lago de datos. Detallando los pasos realizados en cada una de ellas.

Teniendo como destino el realizar un lago de datos de extremo a extremo, incluyendo el análisis del origen del dato y del resultado una vez finalizado todo el procesamiento.

1.5.2-Enfoque metodológico

El método utilizado para la realización de este proyecto está basado en el método Lean (#1.12). Utilizando los preceptos de mejora continua y crecimiento iterativo. En cada sección del desarrollo primero realizaremos un prototipo, lo evaluaremos y a continuación lo extrapolamos al resto de los desarrollos en la sección.

Basándonos a su vez en múltiples reuniones con el tutor para mostrarle los avances e ir realizando ajustes sobre el proceso. Afinando de forma iterativa el proceso.

1.6-Planificación

De cara a la planificación, hemos de abarcar cada entorno con el que vamos a trabajar como un mini proyecto en sí mismo. Con instalación, configuración y desarrollo para cada uno. Siendo además revisitables a medida que avanzamos para realizar modificaciones y ajustes requeridos en los pasos posteriores. Identificando las siguientes áreas:

- Configuración de entorno
- Entrada de dato
- Configuración de entorno de almacenamiento
- Procesado de dato
- Visualización del dato
- Análisis

Disponiendo de un tiempo de desarrollo de 7-8 semanas según los tiempos marcados por la UOC para llevar a cabo el mismo.

(1.gráfico de planificación inicial del proyecto junto con planificación real del mismo)

1.6.1-Seguimiento planteado

Siendo este un TFM enfocado como un proyecto laboral, se ha optado por un enfoque laboral en el seguimiento.
Envío de correos a medida que se avanza y un punto de control semanal en que el se actualiza el estado del proyecto mediante videoconferencia. Se comparten en él los avances con el tutor, se comentan nuevos pasos a seguir, se comentan dificultades y se intenta acomodar el desarrollo a las necesidades planteadas por el tutor. Si es necesario se modifican los hitos establecidos o se añaden nuevos.

1.7-Sumario de productos obtenidos

Con la elaboración de este proyecto se plantea elaborar una herramienta que permita el análisis de la relación entre contaminación y meteorología , acercándonos a la posibilidad de dar una respuesta predictiva.

Podemos decir que en lugar de herramienta hemos creado una suite, ya que está conformada a partir de múltiples herramientas relacionadas con tal de dar una respuesta a la necesidad planteada.

Para ello se ha optado por la elaboración de un lago de datos de extremo a extremo. Desde el descubrimiento e ingesta de fuentes de datos hasta la presentación en informes interactivos para el usuario final. Cada uno de los componentes tiene entidad en sí mismo como para ser analizado al detalle, por eso optamos por sumarizarlos para aportar la visión global. Aún más se detallan los pasos seguidos a lo largo del TFM para facilitar la reproducción en cualquier entorno laboral.

Siendo el objetivo el poder ser utilizado y ampliado en un futuro, se ha realizado el proyecto con el enfoque prioritario de hacerlo lo más configurable posible, parametrizando todo dato requerido para poder ampliarlo o modificarlo en un futuro.

1.8-Breve descripción del resto de capítulos de la memoria.

Describiremos muy rápidamente los diferentes capítulos presentes en esta memoria.

2. Introducción al dato
    - Descripción de todos los datos que se van a procesar (contaminantes, climatología, listado de municipios).
3. Entornos
    - Descripción de los diferentes entornos y herramientas que van a ser utilizados en el proyecto. Justificación de los mismos y explicación de otras herramientas disponibles.
4. Entorno Docker
    - Descripción de la herramienta que nos permite desplegar la red de servidores y configuración.
5. Entorno de entrada de datos
    - Descripción de la herramienta NIFI, su configuración y cómo se utiliza para ingestar datos. Descripción de las fuentes de datos disponibles.
6. Entorno de almacenamiento
    - Descripción de la herramienta Hadoop, configuración y como se ha implementado.
7. Entorno de procesamiento
    - Descripción de la herramienta HIVE, configuración y elaboración de consultas.
8. Entorno de visualización
    - Descripción de herramienta Power BI, configuración y diferentes informes realizados
9. Resultado
    - Análisis pormenorizado de los resultados obtenidos al establecer la relación de datos.
10. Conclusiones
    - Conclusiones extraídas ante la finalización del proyecto.
11. Evoluciones futuras
    - Listado de posibles modificaciones a realizar tras la finalización del proyecto.
12. Dificultades halladas
    - Listado de todas las dificultades mayores halladas durante el desarrollo del proyecto.
13. Agradecimientos
14. Glosario

15. Referencias
16. Anexos

2-Introducción al dato

En este apartado se realizará una breve introducción a los datos que se van a procesar durante la realización de este proyecto, para acercar la realidad de los mismos al lector y que se comprenda que magnitudes van a ser procesadas.

No vamos a entrar en el detalle de los orígenes de captación de los mismos que será procesado durante la ingesta. Esto será explicado en el apartado 4.1 (Origen del dato).

Se deben tomar estos datos que se van procesar como parte de la prueba de concepto que es este TFM. Si el usuario, que sigue los pasos indicados en el TFM, quiere procesar más, menos o elementos diferentes, solo debe variar los elementos indicados por parámetro. Pasando a ser procesados estos de forma automática.

Hay que tener en cuenta que no todos los municipios miden todos los factores. Sus sensores están configurados para leer solo unos pocos contaminantes o factores climatológicos. Se ha intentado buscar, de cara a la diversidad del dato, municipios que incluyan el máximo de los datos tenidos en cuenta.

Los 3 grandes bloques de datos que procesaremos serán: Contaminantes, Factores climatológicos y Municipios.

2.1 - Introducción contaminantes.

Los contaminantes que van a ser considerados en este proyecto son: CO, NO, PM2.5 y PM10.

- CO
    - Un gas incoloro e inodoro generado por la combustión de combustibles basados en el carbono (madera, petróleo, carbón). Siendo la fuente predominante los vehículos de transporte.
    - La exposición dificulta la absorción de oxígeno, impidiendo a los glóbulos rojos la captación de oxígeno. En bajas dosis causa dificultad respiratoria, mareos y cansancio. En altas dosis es letal.
- NO
    - Un gas rojizo marrón soluble en agua con fuertes propiedades oxidantes. Su origen suele ser la combustión a alta temperatura de combustibles, como los usados en calefacción, transporte e industria. Es un gas considerado precursor del Ozono.
    - La exposición al gas causa irritación de las vías aéreas y complica cualquier enfermedad respiratoria. Fuertemente asociado al asma.
- PM2.5 y PM10

- Conjunto de partículas inhalables que agrupa sulfatos, nitratos, amonio, sales, carbón, polvo mineral o incluso agua.
- Se distingue por su diámetro aerodinámico entre las 2.5 µm y 10 µm. Las partículas mayores se originan del polen y el producto de la erosión a raíz de la erosión natural, minado y agricultura. Las partículas menores provienen de la combustión y reacciones químicas entre gases.
- Son partículas capaces de entrar a través de los pulmones en el flujo sanguíneo, pudiendo causar isquemias cardiacas, cerebrales o problemas respiratorios. Incluso sin llegar a causar estos problemas agudos, se considera que la exposición a estas partículas aumenta la morbilidad. Siendo apuntada como una causa real de cáncer de pulmón.

Para tener información más detallada se puede consultar la página de la Organización Mundial de la Salud (WHO) acerca de los tipos de contaminantes (#2.1)

2.1.1 - Límites de contaminantes

De cara al análisis realizado, se deben establecer unos límites para la contaminación tanto diaria como anual, siendo la superación de estos algo nocivo para el ser humano. Marcando el mantener los valores por debajo como el objetivo de cada municipio investigado.

A partir del BOE (real decreto de 01/2023) (#2.2) se establecen una serie de límites para cada contaminante estudiado. Teniendo una réplica también en la Organización Mundial de la Salud.

Siendo los valores establecidos por la OMS mucho más restrictivos y completos, **hemos decidido usar los valores de la OMS para el establecimiento de nuestros KPI**. Teniendo en cuenta que todo el proyecto se ha realizado de forma configurable. En caso de decidir utilizar los límites impuestos por el BOE, o unos diferentes, sería sencillo reemplazar unos valores de límite por otros a partir del reemplazo de ficheros de configuración o añadido de nuevos.

	OMS		**BOE**	
	Límite diario	**Límite anual**	**Límite diario**	**Límite anual**
CO	4		10	
NO	25	10		40
PM2.5	15	5		25
PM10	45	15	50	

(límites para cada contaminante establecidos por la OMS y el BOE)

Desgraciadamente no se ha encontrado ningún tipo de límite establecido por hora, salvo en el caso de NO, por lo que se ha decidido utilizar los límites diarios también para el realizado del control horario.

2.2 - Introducción factores climatológicos.

En este proyecto vamos a tener en consideración una serie reducida de factores climatológicos para analizar su relación con la contaminación.
Estos datos son obtenidos de la red de estaciones meteorológicas automáticas (XEMA) que forman parte del "Servei Meteorologic de Catalunya - Meteocat".

- Presión atmosférica
    - El peso de la atmósfera sobre nosotros, medido en hectoPascales (hPa). Una baja presión indica inestabilidad atmosférica y una alta presión implica una baja variabilidad climatológica.
- Humedad
    - La medida del vapor de agua en la atmósfera. Medida en porcentaje
- Temperatura
    - La magnitud de calor medible por un termómetro, en este caso, grados Celsius.
- Velocidad del viento (a 1m de altura)
    - Velocidad en metros por segundo del viento medido. Este puede ser medido a diferentes alturas para percibir las diferencias (1m, 2m, 6m y 10m).
    - Hemos optado por la medición a 1m debido a observar que era la más contemplada en un gran número de municipios. Sería más óptima la medida a 2m coincidiendo con los sensores de contaminación, pero ante la escasez de esta medición, se ha optado por la más cercana.
- Dirección del viento (a 1m de altura)
    - Dirección de la racha de viento medida en grados de 0 a 360.
    - Nos será muy útil para observar si hay relación en la dirección del viento para la evolución de la contaminación. Por ejemplo, si el viento proviene de una ciudad muy poblada/contaminada y arrastra ese factor contaminante.
- Precipitación en mm
    - Control de la precipitación registrada en mm.
    - Nos servirá para observar si la presencia de la lluvia y su intensidad afecta a la contaminación.

La relación de los factores posibles a analizar depende únicamente de la red de sensores XEMA. Pudiendo ser consultada en la página misma del metadato (#2.3)

2.3 - Introducción a los municipios escogidos.

Teniendo establecidos los elementos que estudiaremos de contaminación y meteorología debemos ahora aplicarlos a un ámbito geográfico. Hemos decidido enfocarnos en un conjunto de 3 ciudades.
En caso de querer establecer más ciudades simplemente debemos añadirlos a los ficheros de configuración y asignar una relación con las estaciones climatológicas y de contaminación pertinentes. A partir de eso la carga se realizará de forma automática.

- Manlleu 
    - Municipio de interior de la provincia de Barcelona, ubicado en la comarca de Osona. Con una población de superior a 21k y una superficie de 17,2 km2. que le otorga una densidad de 1633 hab por km2.
    - Sus coordenadas son 42°00′00″N 2°17′01″E
    - Es objeto de estudio por tener unos valores anormalmente altos para su población y la sospecha de recibir la contaminación de ciudades cercanas (especialmente Barcelona).
- Mataró 
    - Municipio de la costa de la provincia de Barcelona, ubicado en la comarca del Maresme. Con una población de superior a 129k y una superficie de 22,53 km2. que le otorga una densidad de 5598 hab por km2.
    - Sus coordenadas son 41°32′00″N 2°27′00″E
    - Objeto de estudio por su alta densidad de población en comparación con otros municipios.
- Vilanova i la Geltrú 
    - Municipio de la costa de la provincia de Barcelona, ubicado en la comarca del Garraf . Con una población de casi 69k y una superficie de 34 km 2 que le otorga una densidad de 1944 habitantes por km2.
    - Sus coordenadas son 41°13′27″N 1°43′33″E
    - Objeto de estudio por ser una población costera en el sur del vértice creado por los 3 municipios estudiados con unos sensores que aportan datos muy completos.

Posicionamiento de los 3 municipios:

(2.ubicación en mapa de los municipios evaluados)

3-Entornos

En este capítulo introduciremos los diferentes entornos que se han generado para la realización de este proyecto. Las necesidades que lo han impulsado, las diferentes herramientas posibles y justificación de las diferentes herramientas escogidas.

3.1-Contexto

Para el análisis que se tiene planteado se tiene que hacer frente diferentes condiciones. Tratando solamente el dato:

- **Gran volumen de datos**
    - Solamente un municipio por día genera unos 300 registros nuevos en nuestra prueba de concepto (acumulando meteorología, contaminación y predicción) sin realizar ningún tipo de procesamiento. Una vez procesados los datos la cifra escala considerablemente.
    - Aumentando el número de municipios y contaminantes se observa que el volumen de datos diarios es elevado, teniendo en cuenta que la intención es mantener una ventana de dos años de datos, estamos hablando de un gran peso en registros, que influye en el procesamiento posterior y visualización.
- **Alta variabilidad de los datos**
    - Realizando la relación municipios, contaminantes y condiciones meteorológicas nos encontramos con una muy alta variabilidad del mismo dato.
    - Esto a su vez se complementa con el hecho sobre los diferentes orígenes de datos. Aún perteneciendo a la misma red aplicaciones (meteocat-gencat), cada consulta genera datos en diferentes formatos y estructuras, no guardando relación entre sí. Compartiendo únicamente el poderse acceder al dato en formato json
- **Necesidad de alta disponibilidad**
    - Para poder aportar valor al usuario es necesario que los datos puedan ser ofertados lo antes posible. Una vez publicados deben ser ingestados y procesados rápidamente.
    - Se debe aligerar al usuario de todo posprocesamiento del dato, intentando ofrecerle el dato lo más depurado y trabajado posible, por si se cambiase la herramienta final de visualización, que este fuera lo menos traumático posible.

Teniendo en cuenta el objetivo:

- Debe desarrollarse una herramienta fácilmente escalable, donde no sea traumático introducir elementos nuevos, tanto para acelerar procesos como para aumentar espacio o incluso añadir nuevas funcionalidades.
- Es importante tener en cuenta que las herramientas evolucionan y que en el futuro puede que aparezcan componentes más útiles y eficaces. Debemos poder reemplazar un componente del lago por otro sin que sea algo traumático.
- Se debe tener en cuenta el alto espacio que puede ser necesario para almacenar los datos de cara a futuro.

- Es importante la alta configurabilidad de los elementos que se utilicen, y su facilidad. Para que en un futuro si cambian las necesidades de análisis, sea simple el poder modificar unos elementos de análisis por otros sin producirse un drama en la modificación.
- Se intentará reducir costes utilizando todas las herramientas gratuitas posibles y caso de necesitar una herramienta de pago, sugerir la más barata posible.

Teniendo esto en cuenta rápidamente nos damos cuenta de que nos hallamos ante un escenario de Big Data.

Para poder abarcar todas estas condiciones y poder alcanzar los objetivos fijados, hemos optado por la elaboración de un lago de datos de extremo a extremo como herramienta para la realización de este TFM.

La elaboración de una suite que cubra todas las necesidades actuales del usuario y que pueda adaptarse fácilmente a nuevas necesidades. Siguiendo el espíritu del proyecto como TFP (Trabajo profesionalizador) que desarrolle una herramienta funcional para el usuario y que sea fácilmente reproducible en otro entorno.

Cubriremos desde la ingesta de datos a la visualización del mismo en una serie de informes.

3.2- Introducción técnica

Una vez tenido en cuenta los requisitos y limitaciones autoimpuestas del proyecto debemos tener en cuenta las diferentes capas que requerimos para elaborar un lago de datos. Las introduciremos a continuación y les dedicaremos apartados específicos a cada una de ellas:

- Ingesta de datos
- Almacenamiento del dato
- Procesado del dato
- Visualización/Análisis del dato

Cada apartado mencionado puede ser gestionado por una o varias aplicaciones diferentes, según las necesidades específicas del proyecto..

Debido a las limitaciones que nos impone el tiempo disponible para la realización del proyecto, hemos optado por una arquitectura simple con una red de diferentes servidores para cada una de las necesidades/áreas que tenemos dentro de una misma máquina (el propio ordenador personal). No sin antes haber estudiado el mercado y las diferentes opciones posibles para el desarrollo.

3.3-Descripción del mercado, estudio de diferentes opciones posibles

La herramienta más cercana a la intención del proyecto se registra en el proyecto Calíope (#3.1) donde se realizó hasta el 2022 un seguimiento de los datos meteorológicos y de contaminación. Que mediante el análisis de los datos aportados por sensores llegaba a realizar predicciones de la contaminación (emisiones y calidad del aire) y meteorología a nivel nacional y europeo. Elaborando

modelos predictivos al respecto. Pero no realizando un cuadro que permitiera cruzar ambas referencias.

No habiendo encontrado una herramienta o estudio similar que pueda cubrir al completo las necesidades planteadas tenemos que observar las herramientas que puedan cubrir nuestras necesidades individuales e integrarlas de la forma más armoniosa posible.

Para la realización del siguiente estudio hemos partido de experiencia personal y consultas en blogs de especialistas en la materia (#3.2 y #3.3).

3.3.1-Herramientas para la ingesta

Para la fase de ingesta de datos se nos plantean múltiples opciones, de entre las variadas herramientas ETL que tenemos en el mercado. Siendo las principales las que nos llaman más la atención:

- Pentaho 

    - Herramienta desarrollada por Hitachi para la creación de ETL que pueden llegar a ser muy elaborados. Su potencial reside en el procesamiento de ficheros cargados y consulta a bbdds.
- Azure Data Factory 

    - Herramienta de pago para la orquestación facilitada por Azure que permite crear flujos de creación, modificación y carga de datos de manera sencilla.
- Apache Airflow 

    - Herramienta desarrollada por Apache para la creación de ETL mediante DAGs.
- Apache Nifi 

    - Herramienta desarrollada por el FBI y posteriormente liberada que permite infinidad de posibilidades mediante el trabajo de datos en flujo. Incorpora una miríada de diferentes procesadores generados por la comunidad dándole una gran flexibilidad.

3.3.2-Herramientas para el almacenamiento y procesado

Para la fase de procesamiento de datos, hemos escogido directamente (por conveniencia con las asignaturas cursadas previamente en el máster y por disponibilidad tecnológica), inclinarnos por trabajar con el entorno Hadoop. Que nos ofrece Pig, Hive y Spark como herramienta de procesado del volumen de datos que planteamos y Ozzie como orquestador de las tareas a realizar internamente.

Nos aportará robustez, escalabilidad para los volúmenes con los que vamos a trabajar, velocidad y este será muy fácilmente disponible e integrable debido a formar parte del proyecto Docker, que ya mencionamos en el apartado anterior de Ingesta de Datos.

Caso de disponer de presupuesto o tener ya necesidades tecnológicas mayores. Podríamos haber optado por opciones cloud como:

- Google cloud.
- Amazon EMR. Dentro de los Amazon Web services.

3.3.3-Herramientas para la visualización

En la etapa de visualización de datos, por comodidad hemos optado entre tres herramientas.

- Jupyter 

:
    - Que nos permitirá en un momento dado del proyecto realizar hojas de cálculo para poder verificar datos analíticos. Muy flexible y simple permitiéndonos programar en Python directamente sobre él.
- Power Bi 
:
    - Ofrecido por Microsoft, nos ofrece una gran potencia y flexibilidad como la herramienta de pago que es. Dándonos la posibilidad de realizar grandes despliegues creando cuadros de mandos que pueden incorporar funciones predictivas.
- Tableau 

:
    - Herramienta orientada a la inteligencia de negocios que nos ofrece también gran flexibilidad, unos precios ajustados y con una posición establecida en el mercado.

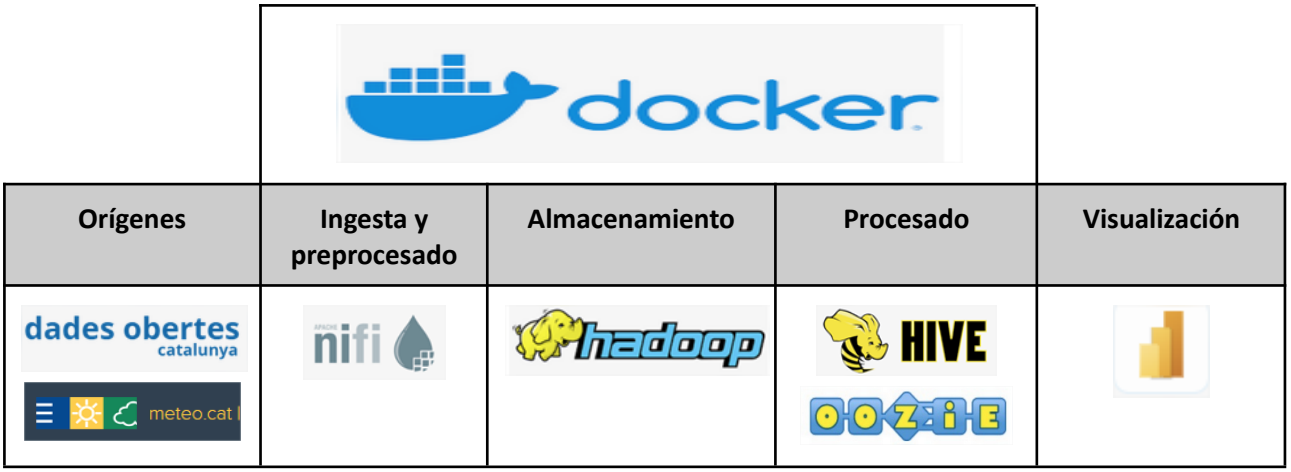
3.4-Descripción de las diferentes herramientas escogidas

Una vez analizadas las diferentes herramientas y atendiendo al criterio de sencillez, familiaridad con la herramienta y velocidad de desarrollo nos hemos inclinado por la siguiente configuración:

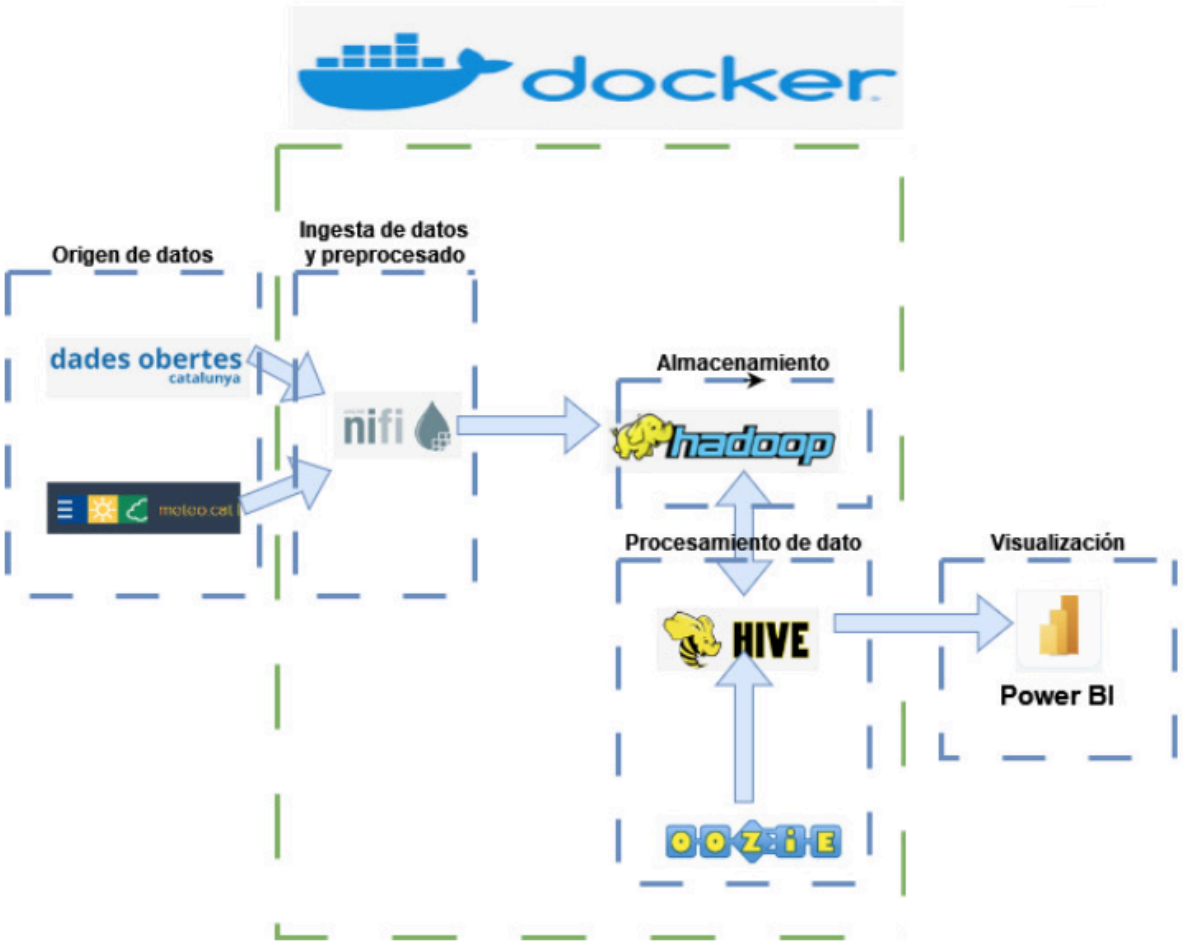
- Ingesta de datos: **Apache Nifi**
    - Debido a su flexibilidad y potencia. Pudiendo incorporar datos almacenados, obtenidos por consulta REST o directamente aportados por sensores.
    - Por familiaridad al haber trabajado con él previamente en el estudio de asignaturas previas.
    - El hecho de estar integrado fácilmente como elemento de Docker.
    - El hecho de su fácil integración con un sistema Hadoop al tener procesadores que permiten gestionar el sistema de ficheros.

- Almacenamiento de datos: **Hadoop**
    - Debido a su facilidad de manejo al permitir el almacenamiento de todo tipo de ficheros sin filtro.
    - El ofrecernos de forma nativa replicación y seguridad en el dato.
    - Su escalabilidad al permitir integrar nuevos nodos si las necesidades aumentan tanto de almacenaje como de seguridad en el fichero almacenado.
    - La amplia documentación y soporte por la comunidad de usuarios que nos facilitará la resolución de cualquier problema en el desarrollo.
- Procesado del dato: **Hive**
    - Debido a su simplicidad al tratar el dato como si fueran tablas SQL. Habilitando un lenguaje de consulta, HiveQL que permite acceder al dato cubriendo nuestras necesidades de consulta.
    - Su altísima velocidad a la hora de consultar el dato (como ejemplo, consultas muy elaboradas que se han ejecutado procesando casi un millón de registros apenas han llevado menos de 2 segundos, en un ordenador personal no muy sobrado de recursos)
    - Es compatible con conexiones odbc y jdbc simplificando el paso de visualización.
    - Trabaja del lado servidor, que es lo que buscamos en todo momento.
- Orquestación del procesado: **Oozie**
    - Una aplicación java web desarrollada por Apache directamente orientada a trabajar contra Hadoop para la elaboración de flujos.
    - Por simplicidad y eficiencia, permitiendo establecer tareas a partir del Hue y relacionar los diferentes procesos estableciendo programaciones de ejecución.
- Visualización de datos: **Power Bi**
    - Debido a su mayor abanico de opciones, facilidad en instalación y conexión.
    - El encontrar una gran comunidad de desarrolladores, aportando respuestas a cualquier problema que apareciera ha pesado en la toma de esta decisión.
    - Después de realizar un estudio se ha observado que sus licencias son ligeramente más baratas que las de la competencia.

Como pegamento que unirá todas estas capas tenemos la herramienta **Docker**. Una plataforma de código abierto hiper configurable que permite integrar una red de servidores en un mismo paraguas. Haciendo que el despliegue, configuración y comunicación entre los mismos sea relativamente sencillo (adelanto, que durante el desarrollo, justamente la configuración fue la parte más difícil).
Puede funcionar tanto bajo Windows como Linux. Por limitación de la máquina huésped, se ha instalado bajo Windows.
Dispone de una gran comunidad detrás que asesora, ayuda con incidencias e incluso aporta distribuciones intentando cubrir necesidades de diferentes usuarios.
Dejándonos esto con un cuadro como el siguiente:



(3.distribución de componentes utilizados para el proyecto)



(4.Gráfico sobre la arquitectura planteada para el proyecto)

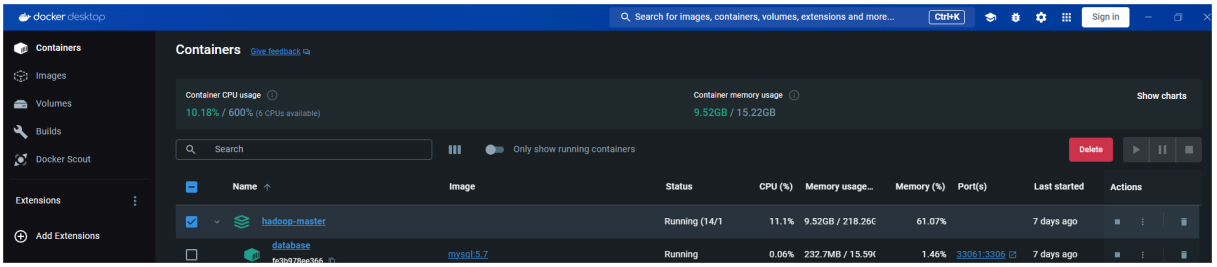
4-Entorno Docker - Red de Servidores

Docker es un proyecto de código abierto utilizado para automatizar el despliegue de aplicaciones que son encapsuladas en contenedores de software. Creada en 2013 por Solomon Hykes.

Se utiliza para desplegar máquinas virtuales (llamadas contenedores) dentro de la plataforma de docker y empaquetar sus dependencias. Puede utilizarse tanto en Linux como en Windows.

En nuestro proyecto ha sido utilizada para realizar todo el despliegue de servidores (incluyendo adicionales que pueden ser utilizados para necesidades futuras, como servidores Spark).

Para ello, se ha descargado de la página principal la herramienta Docker Desktop (#4.1), que nos permite controlar la red de contenedores creada.



(5.Visualización de la herramienta docker utilizada en el proyecto)

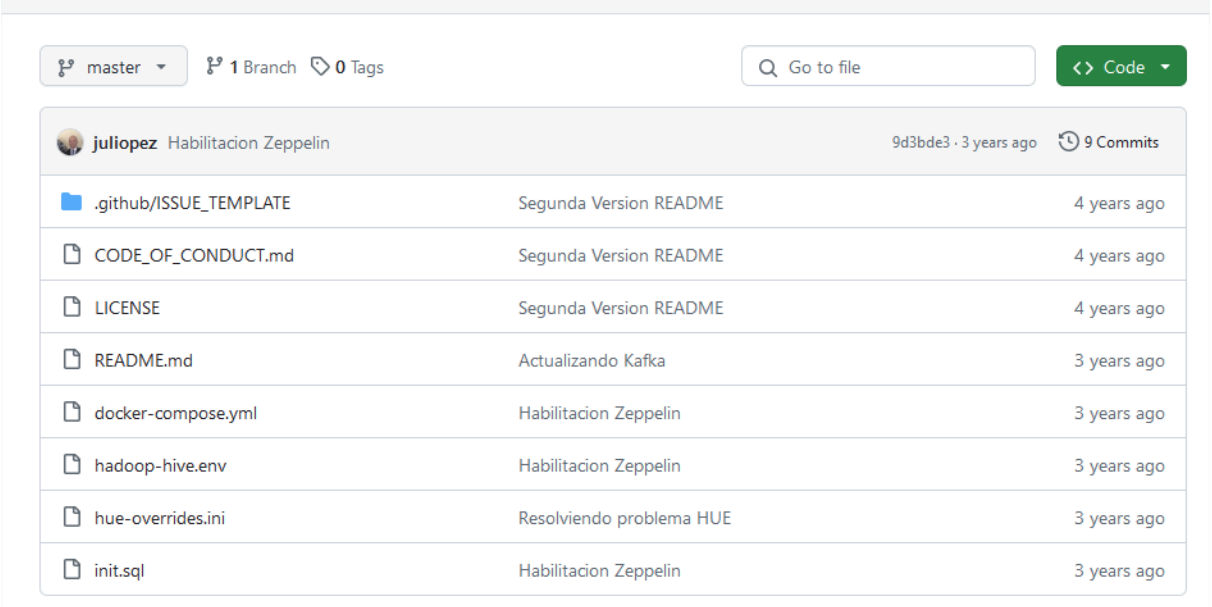
Desde la herramienta se puede controlar la creación de contenedores, creación de carpetas compartidas (llamadas volúmenes) y descarga de nuevas imágenes para crear contenedores.

La descarga de imágenes se puede realizar desde el buscador en la parte superior del desktop o desde la página oficial de docker, una vez creada una cuenta de usuario permite acceder al catálogo de contenedores creados.

Siendo usuarios no experimentados en docker (al menos al inicio del proyecto, las dificultades halladas, descritas en el apartado 11 han cambiado esa situación), lo recomendable es no descargar los contenedores uno a uno y optar por descargar una distribución que contenga ya una red de contenedores creada, que se acomode a nuestras necesidades. Configurando a partir de ese punto.

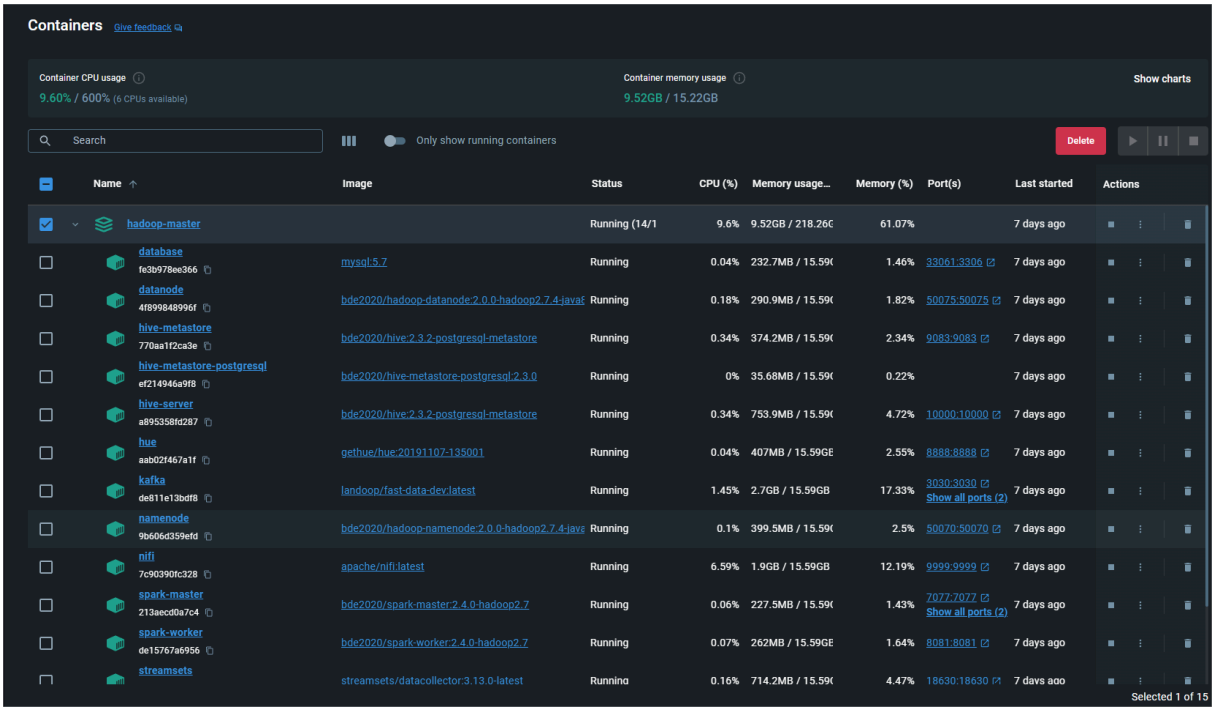
Es decir, descargar una distribución que nos ofrezca funcionalidades de Nifi, Hadoop, Hive, Hue, siendo bienvenidas funciones de Kafka, Spark y Jupyter si es posible.

Después de múltiples pruebas se ha optado por la distribución del especialista Julio Lopez (#4.2). Ofrece nuestros requisitos mínimos, incluyendo además servidores para Kafka y Spark. En este caso, en lugar de descargarlo de la página principal de Docker, lo hemos realizado desde Git. Descargándolo nos aporta adicionalmente ficheros de configuración adicionales (scripts de inicio) para facilitar el despliegue inicial.



(6.Imagen del repositorio git del proyecto big data elaborado por JulioLopez)

Dándonos esto una red de 14 máquinas/contenedores virtuales interconectadas entre sí que pueden correr en nuestro ordenador personal:



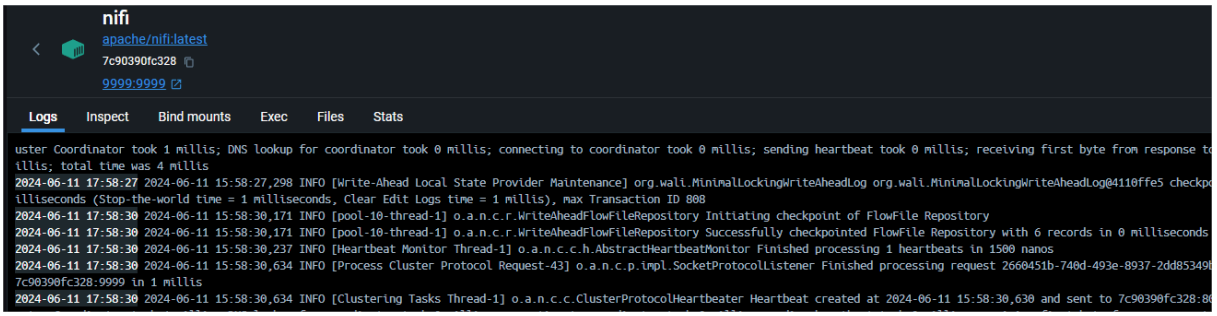
(7.Conjunto de servidores desplegados en la herramienta docker una vez activados)

Todo esto es posible, tras haber realizado la descarga, siguiendo los pasos de configuración necesarios. Es importante consultar el apartado 11.1 donde nos familiarizaremos con todas las dificultades que se pueden generar. Siendo este el apartado más complejo del proyecto (dificultad no esperada sobre el papel).

1. Primero se debe configurar debidamente el fichero descargado docker-compose.yml
    a. Revisando que no hubieran aplicaciones que compartan puerto con las indicadas.

b. Repasando todos los volúmenes creados (sección “volumes” en el fichero). Que estén debidamente mapeados a rutas que existan en nuestra máquina. Punto no opcional, ya que la distribución inicial esta pensada para trabajar en Linux y bajo Windows los volúmenes no se crearían.
c. Creamos volúmenes nuevos para aportar los ficheros de configuración necesarios (hue-overrides.ini, carpetas repositorio para hadoop)
d. Repasando los nombres de las máquinas creadas y dependencias para verificar que no haya ninguna incompatibilidad (nombres duplicados, dependencias que llevan a ninguna parte).
e. Repasando el fichero hue-overrides.ini para una correcta configuración de Hue. Esta parte es obligatoria, al menos con esta distribución.

2. Una vez correctamente configurado, tras la ejecución del comando *docker-compose up* (en la ruta donde este alojado el fichero .yml) Se importarán todos los ficheros necesarios y se levantarán todas las máquinas creadas.
3. Es importante repasar los logs de todos los servidores para verificar que no se ha producido ningún fallo inesperado en la descarga y activación del contenedor. Esto es pulsando dentro del contenedor y accediendo a su sección de logs.



(8.Log del servidor nifi ejecutandose dentro del aplicativo docker)

Una vez realizados estos pasos, tendremos configurado el lienzo sobre el que pretendemos trabajar para el resto de pasos.

5-Entorno Entrada de datos y preprocesamiento- Nifi

La entrada/ingesta de datos es uno de los apartados más importantes en un proceso de creación de lago de datos y en este proyecto ha sido tratado con la debida importancia. Un lago de datos, sin un río que lo nutra se secaría.

De la constante entrada de datos depende que este lago sea funcional. En este caso utilizaremos tres “ríos” para nutrirlo. Tres entradas de datos que serán gestionadas por Nifi y que desembocará los datos preprocesados en la plataforma Hadoop, donde serán almacenados. Dos de ellas son relevantes para el estudio que se realiza en el proyecto y la tercera se utilizará de cara a usos futuros.

Primero procederemos a definir los orígenes de datos con detalle para a continuación detallar los procesos realizados para la inserción.

5.1-Orígen del dato

Para este proyecto tenemos que importar datos sobre la polución provenientes de sensores instalados en municipios y cruzarlos con sensores de datos climatológicos de los mismos. Adicionalmente importamos datos sobre la predicción meteorológica (son para usos futuros de la aplicación, no contemplados en el alcance del proyecto).

Contamos con dos fuentes principales:

- Página de **análisis de transparencia de la generalitat**. El portal "dades obertes de catalunya": https://analisi.transparenciacatalunya.cat/
- Datos publicados por Meteocat (Servicio meteorológico de Catalunya) también a través de la página del análisis de transparencia con la predicción meteorológica: https://static-m.meteo.cat/content/opendata/dadesobertes_pg.json

La limitación que nos ofrece la plataforma son:

- Límite al tamaño de las consultas recuperadas.
    - Esto lo superaremos haciendo consultas incrementales. Solo recuperando los datos existentes trás la última consulta realizada (recuperando la fecha de la última recuperación correcta).
- La necesidad de registrarse para obtener la x-api-key que usaremos en el header de nuestras llamadas para tener un volumen mayor de repuestos y acceso a filtros.

5.1.1-Recuperación de datos de contaminación

Para recuperar los datos sobre la contaminación debemos consultarlos a través de la página del análisis de transparencia. Una vez realizada la consulta de todos los orígenes posibles nos quedamos con Qualitat de l'aire als punts de mesurament automàtics de la Xarxa de Vigilància i Previsió de la Contaminació Atmosfèrica. Este es posible consultarlo a través de llamada REST y aplicar los filtros que consideremos necesarios.

La url de consulta es https://analisi.transparenciacatalunya.cat/resource/tasf-thgu.json y se puede acceder con un método GET.

Los datos recuperados en una llamada vienen en forma de un array de registros que contiene:

- Codi_eoi
    - Código oficial. Concatenación de código de provincia, municipio y estación.
- Nom_estacio
    - Nombre de la cabina de medición. Alfanumérico (100).
- Data
    - Fecha de la lectura en formato yyyy-MM-ddTHH:mmss.sss.
- Magnitud
    - Código numérico del contaminante.
- Contaminant

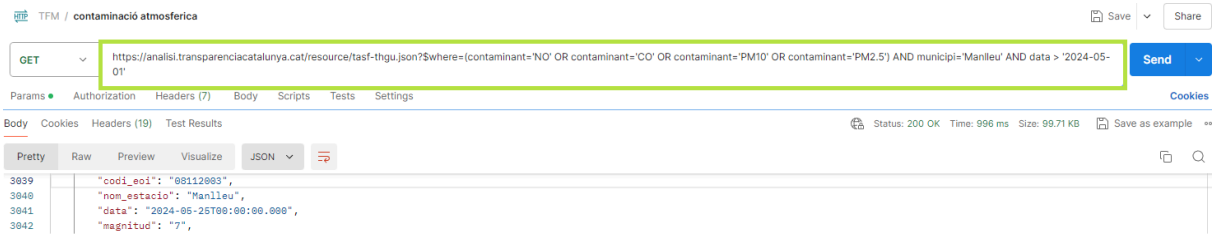
    - Texto del contaminante.
- Unitats
    - Unidades de medida del contaminante. Texto.
- Tipus_estacio
    - Valores posibles: Traffic, Background, Industrial.
- Area Urbana
    - Valores posibles: Urban, Peri-urban,rural.
- Codi Ine
    - Código texto referente al municipio. Concatenación de provincia y municipio.
- Municipi
    - Nombre del municipio donde se halla la estación. Alfanumérico de 100 caracteres.
- Codi Comarca
    - Código referente a la comarca donde se halla la estación.
- Nom Comarca
    - Texto descriptivo de la comarca.
- [Conjunto de hasta 24 columnas que siguen el formato HHh, es decir la hora en dos dígitos con h concatenado]
    - Indica el valor registrado del contaminante para esa hora en cuestión.
- Altitud
    - Expresada en metros respecto a la altura del mar.
- Latitud
    - Expresada en grados decimales según el sistema WGS84.
- Longitud
    - Expresada en grados decimales según el sistema WGS84.

(para más detalle puede ser consultado en : https://analisi.transparenciacatalunya.cat/Medi-Ambient/Qualitat-de-l-aire-als-punts-de-mesurament-autom-t/tasf-thgu/about_data y https://dev.socrata.com/foundry/analisi.transparenciacatalunya.cat/tasf-thgu)

```
3039        "codi_eoi": "08112003",
3040        "nom_estacio": "Manlleu",
3041        "data": "2024-05-25T00:00:00.000"
3042        "magnitud": "7",
3043        "contaminant": "NO",
3044        "unitats": "µg/m3",
3045        "tipus_estacio": "background",
3046        "area_urbana": "suburban",
3047        "codi_ine": "08112",
3048        "municipi": "Manlleu",
3049        "codi_comarca": "24",
3050        "nom_comarca": "Osona",
3051        "h01": "1",
3052        "h02": "2",
3053        "h03": "2",
3054        "h04": "2",
3055        "h05": "2",
3056        "h06": "2",
3057        "h07": "2",
3058        "h08": "3",
3059        "h09": "3",
3060        "h10": "4",
3061        "h11": "4",
3062        "h12": "3",
3063        "h13": "1",
3064        "h14": "1",
3065        "h15": "2",
3066        "h16": "1",
3067        "h17": "1",
3068        "h18": "2",
3069        "h19": "1",
3070        "h20": "1",
3071        "h21": "1",
3072        "h22": "1",
3073        "h23": "1",
3074        "h24": "1",
3075        "altitud": "460",
3076        "latitud": "42.003307",
3077        "longitud": "2.2872992"
```


(9.ejemplo de datos sobre contaminación recuperados en formato .json embellecido)

Para realizar la llamada aplicaremos siempre el filtro respecto a los contaminantes que analizaremos (NO,CO, PM2.5 y PM10) y los municipios evaluados (Mataró, Manlleu y Vilanova i la Geltrú).



(10.Ejemplo de consulta en Postman con filtros establecidos sobre el endpoint de contaminación)

5.1.2-Recuperación de datos meteorológicos

Para recuperar los datos sobre la meteorología en las ventanas temporales que estudiamos debemos consultarlos a través de la página del análisis de transparencia. Una vez realizada la consulta de todos los orígenes posibles nos quedamos con Dades meteorològiques de la XEMA
Este es posible consultarlo a través de llamada REST y aplicar los filtros que consideremos necesarios.

Los datos recuperados en una llamada son un array de registros que contienen

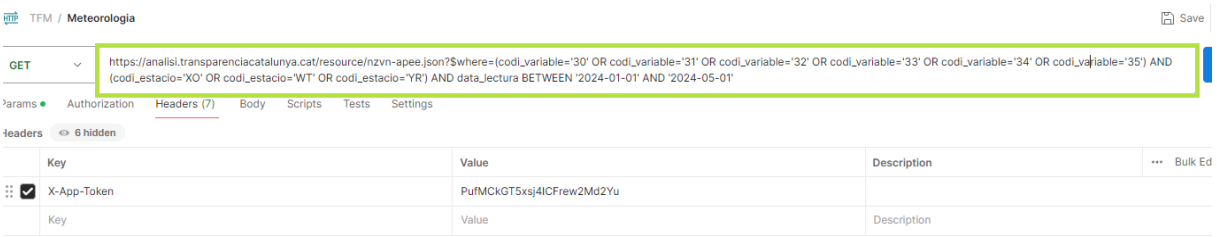
- ID
    - Columna de identificación
- Codi_estacio
    - Código identificador de la estación
- Codi_variable
    - Código de la variable meteorológica
    - Con el propósito del estudio solo tendremos en cuenta:
        - 34 Presión atmosférica
        - 33 Humedad
        - 32 Temperatura
        - 30 Vel. vent (a 1m)
        - 31 Direcció del vent (a 1m)
        - 35 Precipitación en mm
- Data_lectura
    - En formato yyyy-MM-ddThh:MM:ss.sss
- Data_extrem
    - En formato yyyy-MM-ddThh:MM:ss.sss. Corresponde al momento en que se ha registrado el máximo o mínimo.
- Valor_lectura
    - En formato texto que indica un valor decimal
- Codi_estat
    - Indica el estado de validación:
        - (en blanco) - dato sin validar
        - T - en proceso de validación
        - V - Validado
- Codi_base
    - Indica si la medida se realiza de forma horaria (H) o semi horaria (SH)

```
[
    {
        "id": "WT300101240000",
        "codi_estacio": "WT",
        "codi_variable": "30",
        "data_lectura": "2024-01-01T00:00:00.000",
        "valor_lectura": "1.7",
        "codi_base": "SH"
    },
    {
        "id": "WT310101240000",
        "codi_estacio": "WT",
        "codi_variable": "31",
        "data_lectura": "2024-01-01T00:00:00.000",
        "valor_lectura": "3",
        "codi_base": "SH"
    },
```


(11.ejemplo de datos sobre meteorología recuperados en formato .json embellecido)

(para más detalle puede ser consultado en : https://analisi.transparenciacatalunya.cat/Medi-Ambient/Dades-meteorol-giques-de-la-XEMA/nzvn-apee/about_data)

Para realizar la llamada aplicaremos siempre el filtro respecto a los elementos meteorológicos que analizaremos (30,31,32,33,34,35) y las estaciones correspondientes a los municipios estudiados (YR,WT,XO). A su vez incluimos también un filtro por fecha a partir de la que recuperaremos datos (para no recuperar todo sin control).



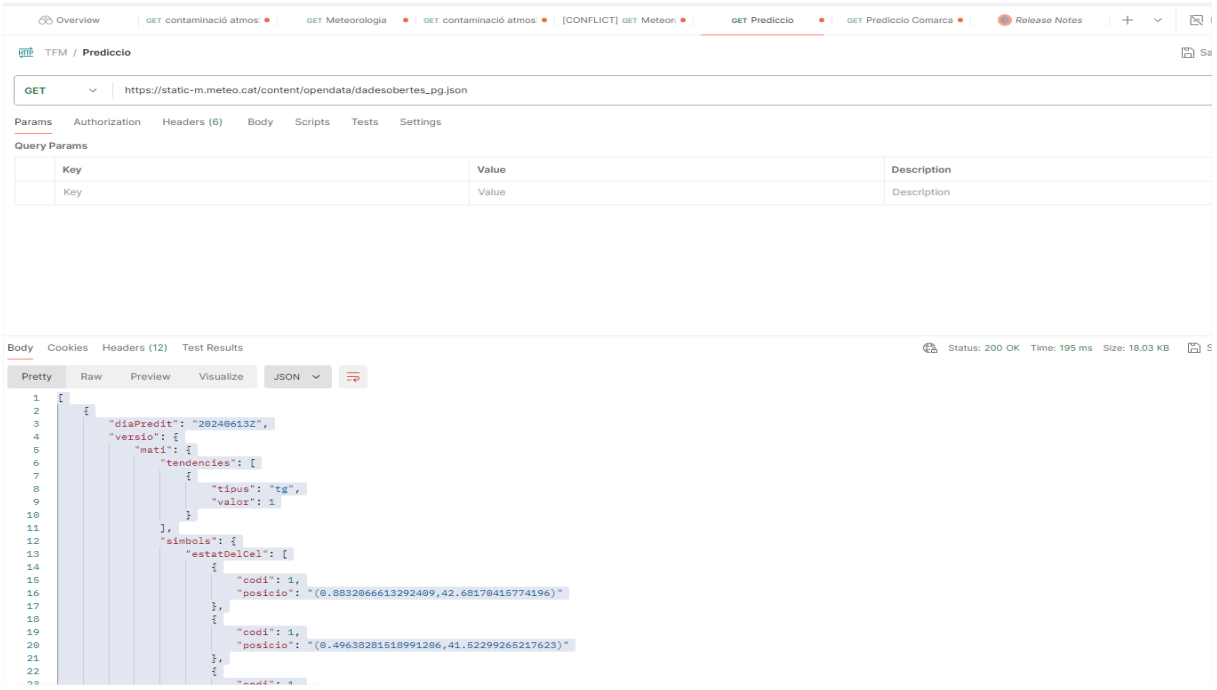
(12.Ejemplo de consulta en Postman con filtros establecidos sobre el endpoint de meteorología)

Se observa una gran diferencia respecto a la fuente anterior. Mientras que los contaminantes crecen en columnas (añadiendo una por hora), en los datos meteorológicos se observa que se aumenta en número de registros. Siendo registros mucho más sencillos y manejables.
El dato de contaminación debe ser “pivotado” de columnas a registros para una gestión más sencilla.

5.1.3-Recuperación de predicciones meteorológicas.

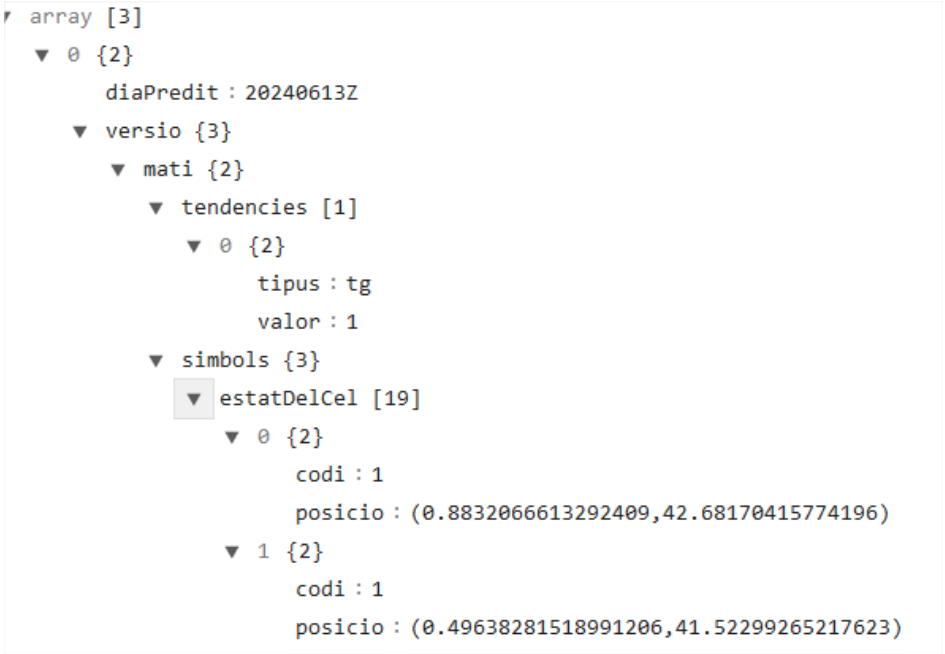
En el caso de la predicción meteorológica este es mucho menos elaborado. Se recupera un fichero json, sin posibilidad de filtro ninguno.
Es el dato completo de la predicción por áreas, publicado 8:30-11:30 y 19:30 con información a 3 días vista únicamente.

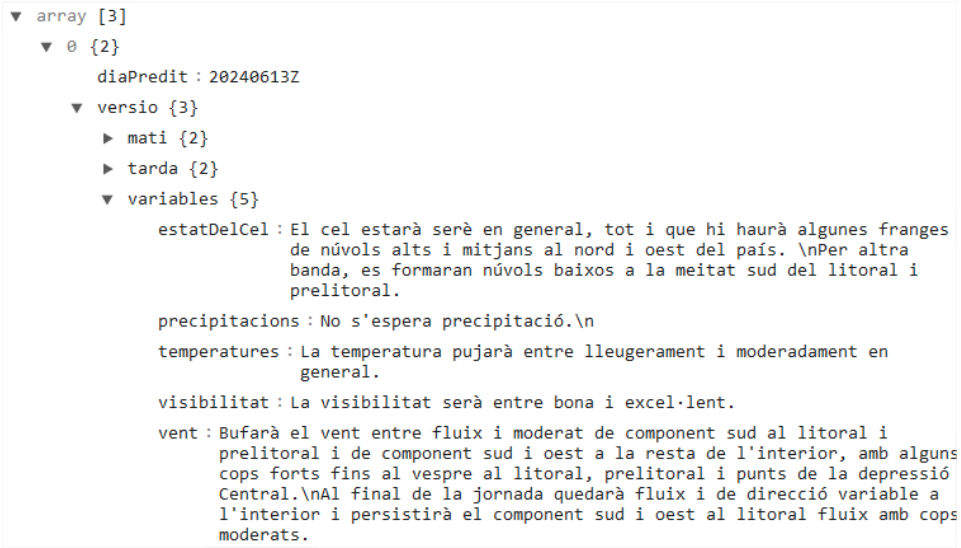
Se debe acceder a la url https://static-m.meteo.cat/content/opendata/dadesobertes_pg.json
Utilizando un método GET. Sin ninguna autenticación



(13.ejemplo de consulta en Postman sobre el endpoint de predicción meteorológica)

Los datos recuperados son:

- Array de 3 posiciones (cada una correspondiente a uno de los 3 días predichos, en orden). Contiene
  - diaPredit: fecha en formato yyyyMMdd
  - versio: un array que contiene en su interior las tendencias para la mañana, la tarde (dentro de las tendencias tenemos un array con el geoposicionamiento, y su variable asociada) y una descripción de las variables en texto plano
  - 



(14.ejemplo de datos recuperados de predicción meteorológica, embellecidos)

Como se observa es un dato bastante sencillo y poco trabajado. Nos ofrece únicamente el estado del cielo, del mar y el viento en formato tendencia (sube/baja). No da mucha posibilidad de juego a posteriori.

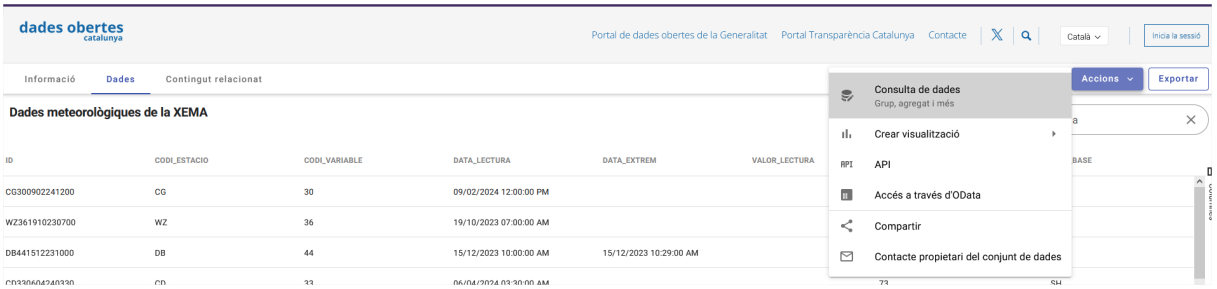
Por este motivo hemos guardado el dato, pero lo hemos enfocado para usos futuros.

5.1.4-Captación dato histórico

Con el desarrollo que detallaremos en el punto 5.2 dejamos preparado el sistema para importar datos frescos del sistema. Pero nuestra intención es trabajar datos de una ventana más grande.
Queremos captar una ventana de datos de 2 años atrás del año actual. En este caso desde el **1 de enero de 2022.**
Mediante llamadas REST esto no es posible por el volumen, supera el máximo autorizado en las transacciones con la página de transparencia.
Para ello directamente creamos vistas en el portal (tanto en polución como en climatología), descargando los datos como .csv y a continuación importarlos en Hadoop manualmente.
Finalizamos el procesado adecuadamente con HIVE.

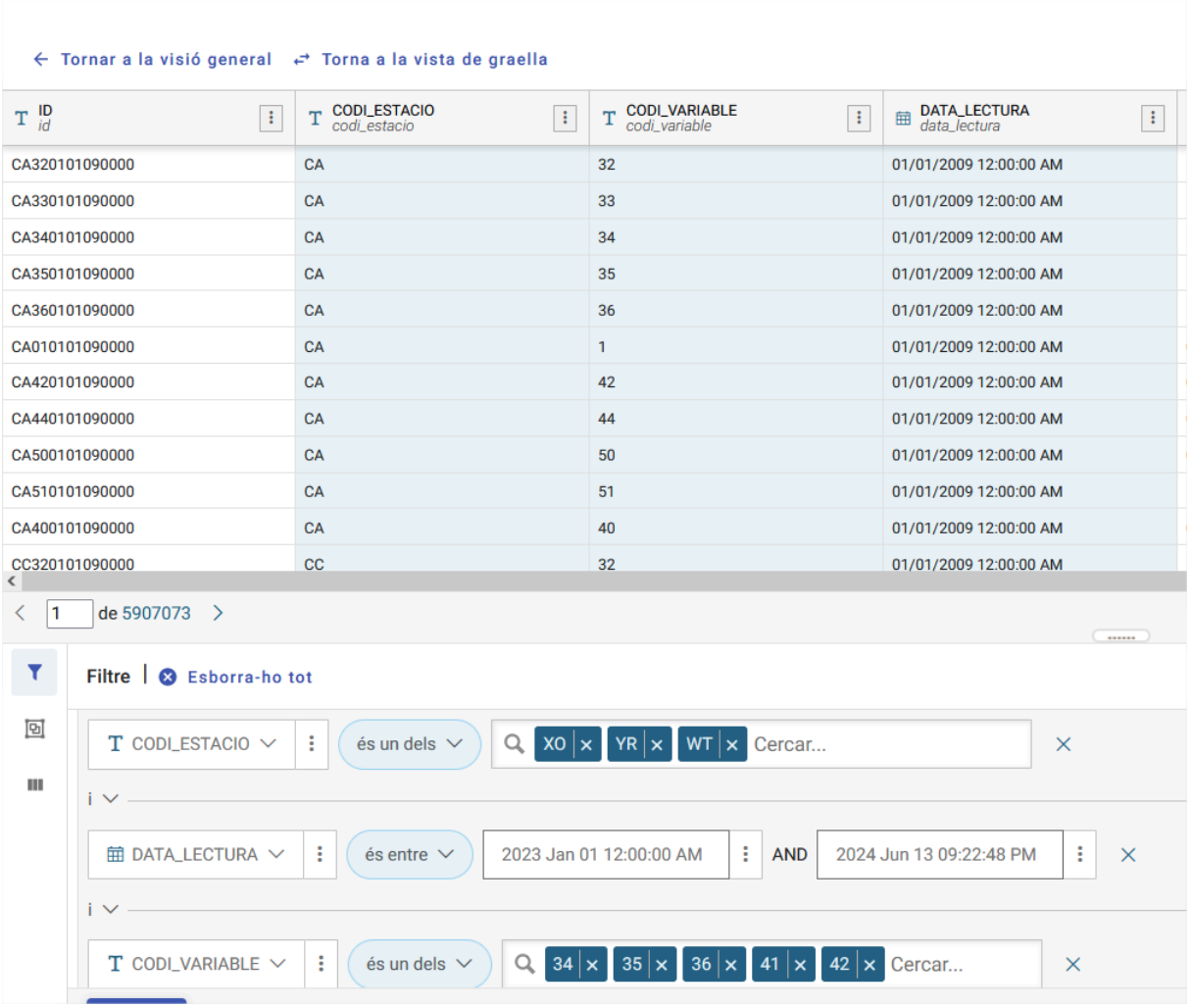
Detallamos los pasos:

- Desde la página deseada (Accions->Consulta de dades). En este caso la de meteorología de la XEMA.



(15.Pantalla de generación de informes en el portal de la XEMA, para recuperar datos históricos)

- Una vez dentro, establecemos los filtros que nos interesen para acotar el dato



(16.Ejemplo de generación de filtrado en la página de elaboración de consultas).

- Una vez filtrado el dato solo debemos pulsar en el botón exportar y nos permitirá descargarlo como fichero .csv (tarda tiempo en procesarlo, hay que tenerlo en cuenta).
    - Con el fichero importado, lo subiremos manualmente al sistema Hadoop con los comandos manualmente. Como ejemplo: (la última carga que realizamos antes de finalizar el desarrollo):
      hdfs dfs -mkdir -p /TFM/Sources/historicMeteo
      hdfs dfs -put /etc/sourceFiles/Dades_meteorol_giques_de_la_XEMA_20240527_format.csv /TFM/Sources/historicMeteo/

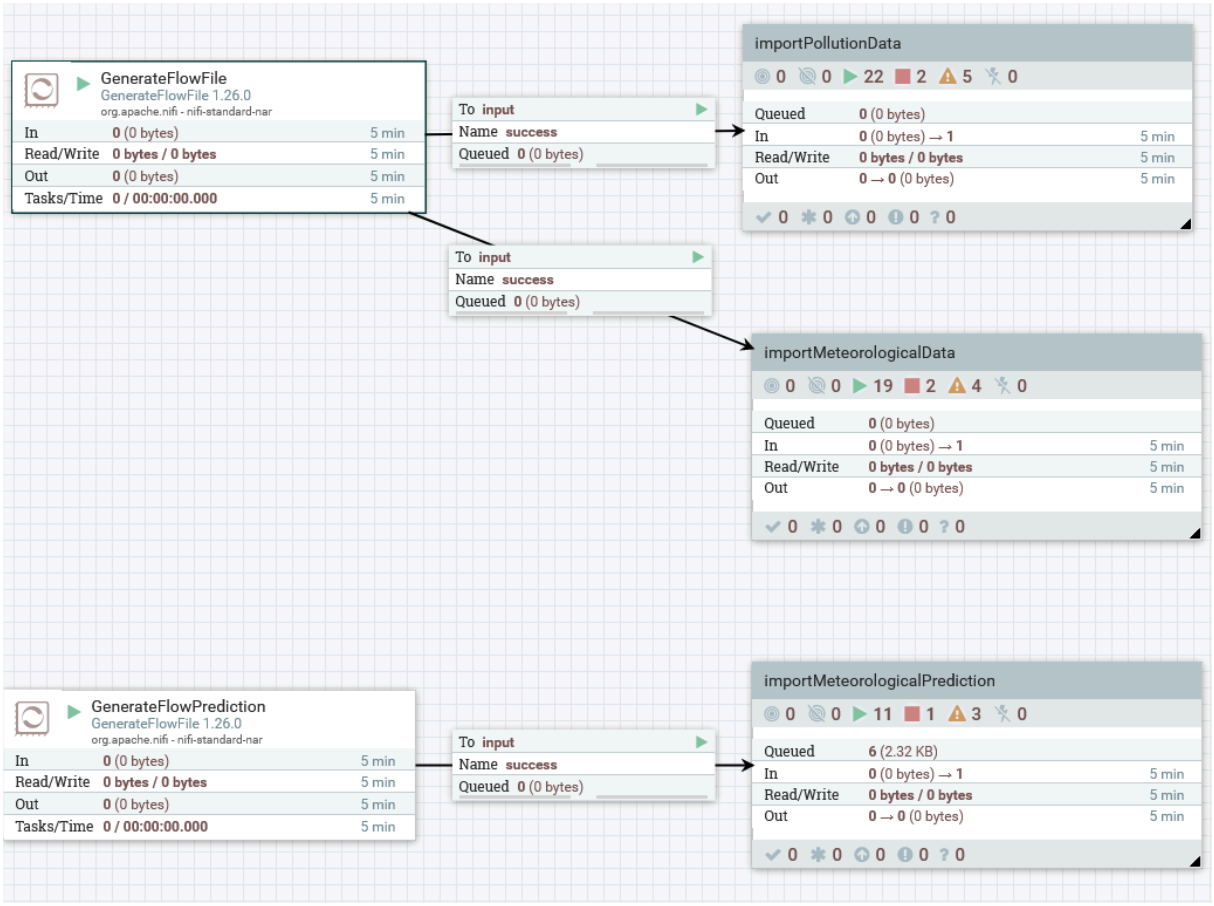
5.2-Captación del dato - NIFI

Para realizar la ingesta utilizaremos la aplicación Apache Nifi, una herramienta que encaja como un guante para este proyecto. Su capacidad de gestionar múltiples flujos, creando disparadores independientes para cada uno de ellos, su gran disponibilidad de procesadores para conectar con variados entornos y una interfaz gráfica muy potente que simplifica el trabajo, desarrollo e interpretación de los flujos creados fácilmente nos hacen favorecer esta herramienta para esta sección del proyecto.

Una ventaja adicional, que usaremos para documentar el proyecto es la posibilidad de observar el dato y cómo evoluciona a partir de cada paso del proceso de ingesta. El hecho de que permita modificar el desarrollo en caliente y trazar los errores con relativa facilidad es un plus añadido.

Para esta sección del proyecto, utilizaremos un docker de Apache Nifi con la versión 1.26.0 de Nifi. Una vez configurado nos permitirá acceder al escritorio de Nifi para crear nuestro proceso.

Una vez dentro del escritorio observamos:



(17.Pantalla principal del proyecto de captación en el escritorio NIFI)

Tres grupos de proceso, activados por procesadores con un procesador activado temporalmente.

- Grupo de procesos para importar contaminantes (importado diariamente a las 11:00)
- Grupo de procesos para importar datos climatológicos (importado diariamente a las 11:00)
- Grupo de procesos para importar predicciones (importado diariamente a las 9:00)

Hay que denotar que para las variables constantes que se usarán en el proceso como:

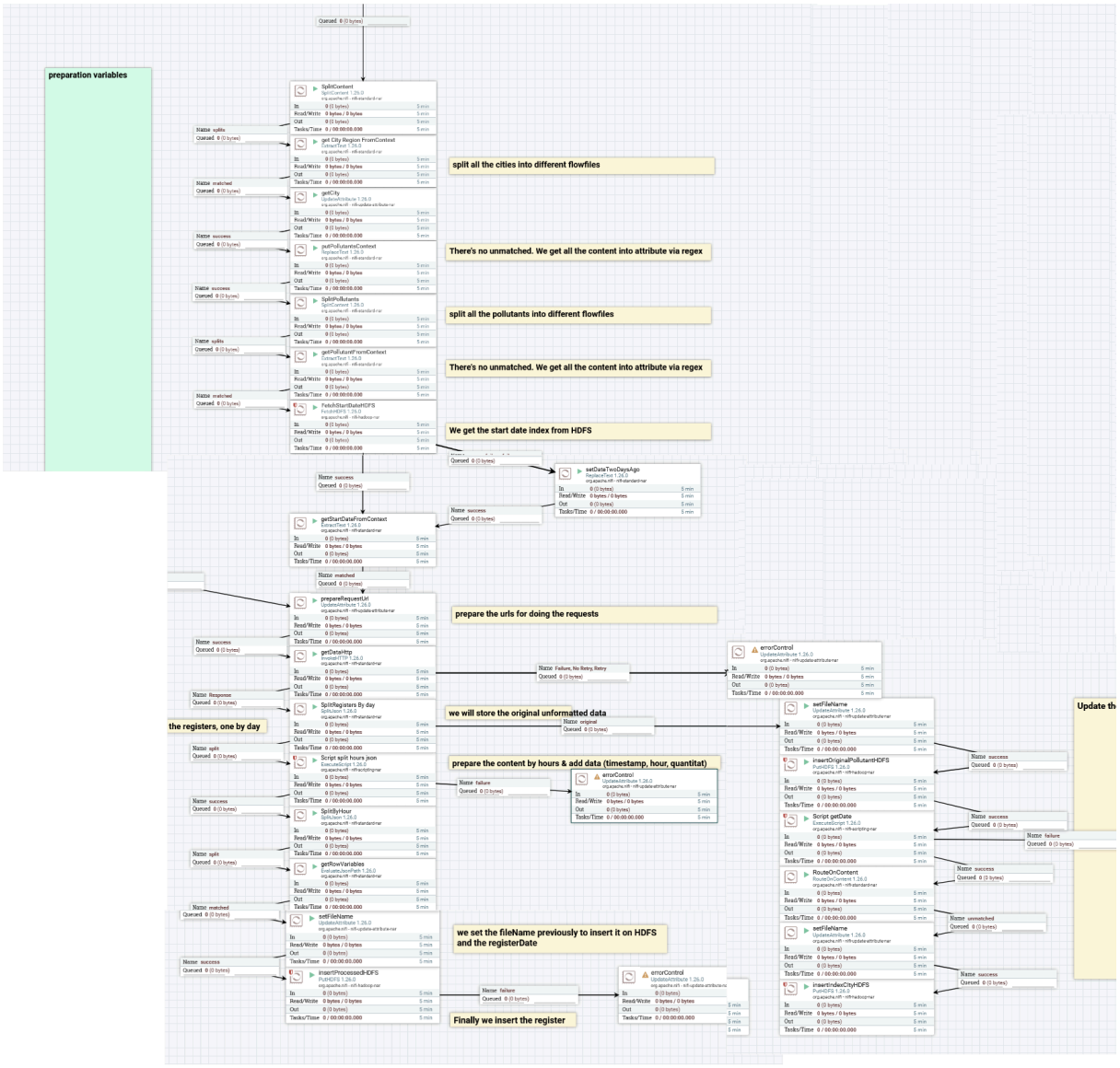
- Límites
- Urls
- Parámetros

Serán almacenados en el contexto de parámetros de Nifi como constantes, en varios contextos que mantendrán herencia para poder manejar y parametrizar a posteriori el proceso, en lugar de repetir urls en cada procesador, por ejemplo.
Estos parámetros serán detallados en el apéndice del documento.

Procederemos a detallar los dos primeros grupos de proceso al ser los principales (el grupo de procesos de predicción es mucho más sencillo y puede ser inferido)

El proceso principal (ImportPollutionData) será detallado en la sección de anexos, siendo el más elaborado y completo. A partir de él se puede inferir el resto.

5.2.1-Captación de contaminación



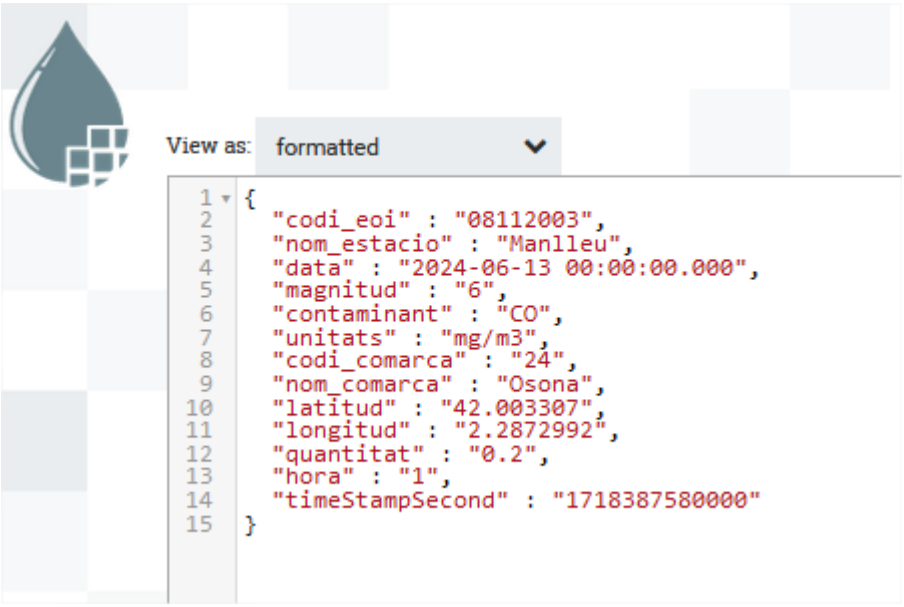
(18.Proceso NIFI de captación de registros de contaminación)

Hemos generado un proceso matriz que capta todos los datos dentro del filtro de fecha de creación, contaminante y municipios creados.
Primero consultará en HDFS cuál ha sido la última consulta exitosa (para obtener el filtro de fecha correcta), reordenará el dato, orientando de creación en aumento de columnas a aumento de registros.
Finalizando con el almacenamiento en HDFS tanto del dato procesado como el original. En una carpeta que permite el almacenamiento sin necesidad de machacar el dato previo.

Detallamos los procesadores utilizados (agrupando por funcionalidad).

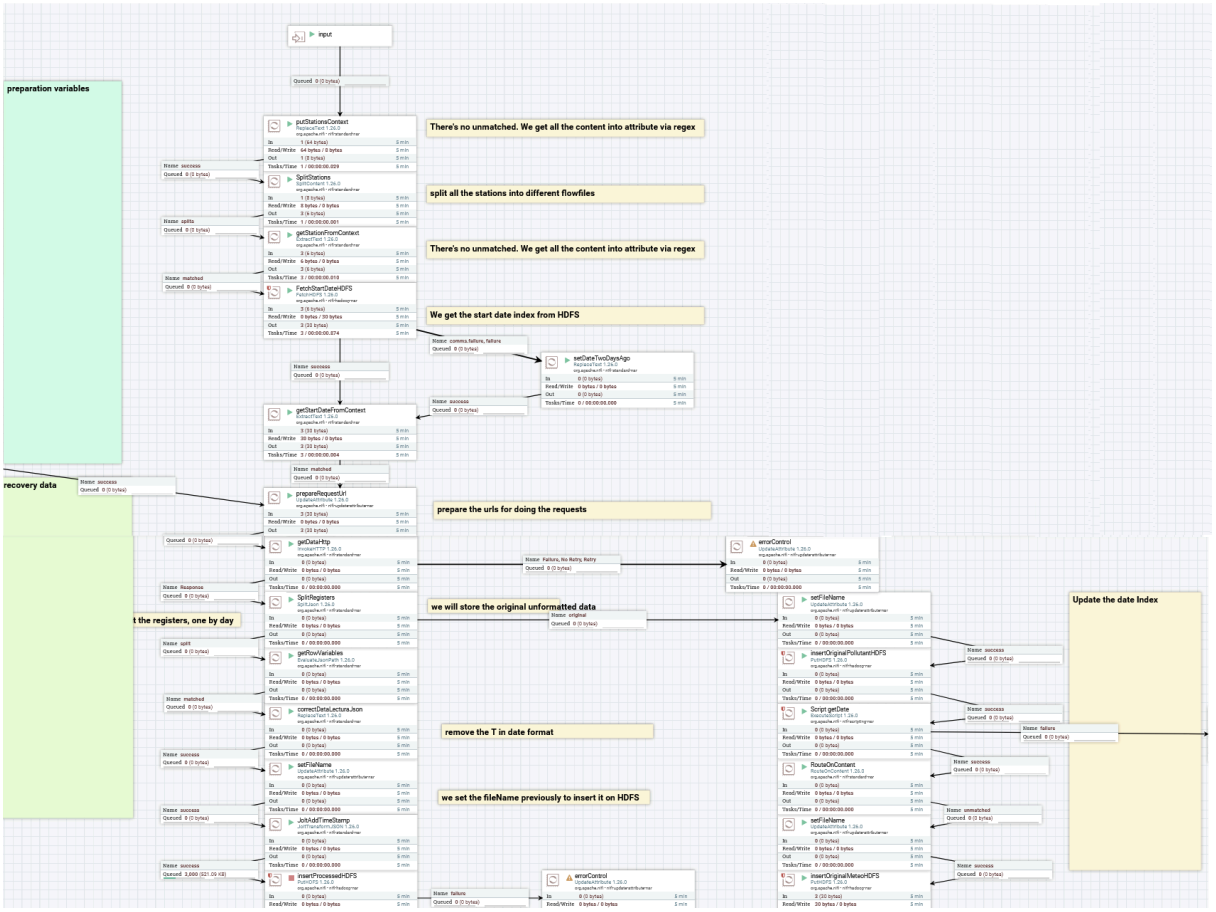
1. ***SplitContent, get City RegionFromContext, getCity***
    - Segregamos el dato recibido del procesador principal (un array de municipios). Creando un hilo por cada municipio.
    - Recuperamos como variable el municipio (para poder usarlo posteriormente).
    - Realizamos un ajuste en el texto recibido para prepararnos para las consultas
2. ***putPollutantContext, SplitPollutants, getPollutantFromContext***
    - Insertamos en el contexto del hilo principal el array de contaminantes. Ya tenemos almacenados los datos de municipios. Ahora queremos realizar un split por contaminante.
    - Segregamos el dato recibido del procesador principal (un array de contaminantes). Creando un hilo por cada municipio y contaminante.
    - Recuperamos como variable el contaminante (para poder usarlo posteriormente).
3. ***FetchStartDateHDFS, setDateTwoDaysAgo, getStartDateFromContext***
    - Primero intentamos recuperar la última fecha procesada de la consulta (está almacenada en un fichero de texto plano en el sistema hdfs). Si existe el fichero, almacenará el valor en el contexto del flujo Nifi.
    - Si este fichero no existiera, o tuviéramos un fallo de comunicación, el procesador setDateTwoDaysAgo, simplemente asignará una fecha al contexto (correspondiente a Today menos dos días).
    - Finalizamos captando la fecha que haya en el contexto, que será utilizada para la consulta REST.
4. **PrepareRequestUrl,GetDataHttp**
    - Preparamos el filtro de consulta que añadiremos a la consulta. Podría realizarse directamente en el siguiente paso, pero es más cómodo y legible hacerse el cálculo en un procesador independiente.
    - Realizamos la llamada GET para obtener todos los datos de polución.
5. **SplitRegisters By Day, SplitHours json, Split By Hour**
    - Realizaremos un split de la respuesta recibida (un array de registros) para crear un registro por día.
    - Mediante script groovy, realizaremos un split de la estructura del registro. Que está orientado en expansión columnar por horas. Pivotaremos el resultado para obtener múltiples registros, uno por cada columna hora.
        - Adicionalmente incorporamos una variable con el tiempo en formato timestamp.
    - A continuación crearemos un flowfile por cada registro nuevo creado, de cara a su almacenamiento en HDFS.

6. **GetRowVariables, SetFileName**
    - Captamos datos del registro actual, de cara a preparar el nombre de fichero
    - Modificamos el nombre de fichero que asigna por defecto NIFI.
        - [código Eoi]_[contaminante]_[fecha de lectura en formato yyyyMMdd_HHmm_HH]
7. **InsertProcessedHDFS**
    - Inserción del fichero en el sistema hdfs, en la carpeta asignada para el procesado de los datos en HIVE de la contaminación.
8. **(rama paralela) SetFileName, InsertOriginalPollutantHDFS**
    - En esta rama se insertarán los datos originales sin procesar, ante posibles usos futuros.
    - Se asigna un nombre de fichero (para machacar el nombre aleatorio generado por Nifi. el formato es [municipi][fecha en formato: yyyy_MM_dd]_[fecha fin en formato: yyyy_MM_dd]
    - Se inserta el fichero sin modificar en el sistema hdfs, en la carpeta asignada para el almacenamiento de datos históricos.
9. **(rama paralela) Script getDate, RouteOnContent, SetFileName, InsertIndexCityHDFS**
    - A continuación almacenaremos la fecha procesada en el sistema HDFS. Habrá un fichero por municipio y contaminante. Almacenando la fecha más alta procesada.
    - Mediante script groovy se recorre la respuesta, almacenando la fecha más alta para cada contaminante y municipio.
    - Se prepara un nombre de fichero.
        - Un valor fijo, ya que machacamos el fichero de parámetros existente en el sistema. Formato [municipio]_[contaminante]
    - Almacenaremos en el sistema hdfs el parámetro para usos futuros.



(19.Ejemplo de registro de datos captados y procesados de contaminación)

5.2.2-Captación de datos meteorológicos



(20.Proceso NIFI de captación de registros de meteorología)

Hemos generado un proceso matriz que capta todos los datos dentro del filtro de fecha, estación y condición climatológica creado.
Primero consultará en HDFS cuál ha sido la última consulta exitosa (para obtener el filtro de fecha correcta). Tras la consulta segregará el dato en múltiples registros, por fecha/hora, condición climatológica y estación.
Finalizando con el almacenamiento en HDFS tanto del dato procesado como el original. En una carpeta que permite el almacenamiento sin necesidad de machacar el dato previo.

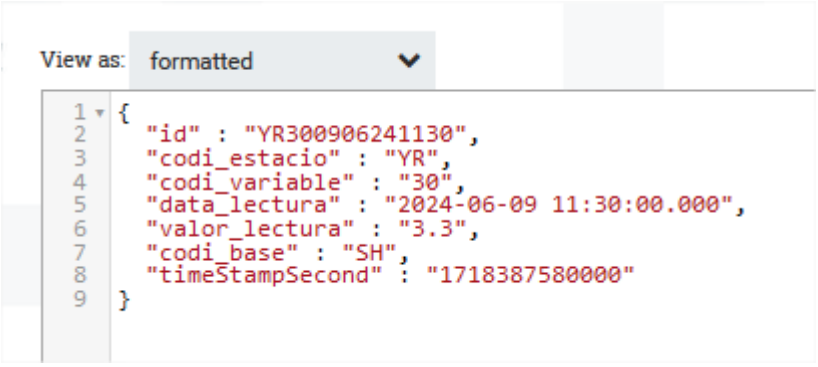
Detallamos los procesadores utilizados (agrupando por funcionalidad).

1. ***PutStationsContext, SplitStations, get Station FromContext,***
    a. Insertamos la lista de estaciones con las que trabajaremos (extraída de los parámetros) al contexto del flujo Nifi
    b. Segregamos el dato recibido del procesador principal (un array de estaciones). Creando un hilo por cada estación.
    c. Recuperamos como dato la estación meteorológica del contexto (para poder usarlo posteriormente).
    d. Realizamos un ajuste en el texto recibido para prepararnos para las consultas
2. ***FetchStartDateHDFS, setDateTwoDaysAgo, getStartDateFromContext***

a. Primero intentamos recuperar la última fecha procesada de la consulta (está almacenada en un fichero de texto plano en el sistema hdfs). Si existe el fichero, almacenará el valor en el contexto del flujo Nifi.
b. Si este fichero no existiera, o tuviéramos un fallo de comunicación, el procesador setDateTwoDaysAgo, simplemente asignará una fecha al contexto (correspondiente a Today menos dos días).
c. Finalizamos captando la fecha que haya en el contexto, que será utilizada para la consulta REST.

3. **PrepareRequestUrl,GetDataHttp**
    a. Preparamos el filtro de consulta que añadiremos a la consulta. Podría realizarse directamente en el siguiente paso, pero es más cómodo y legible hacerse el cálculo en un procesador independiente.
    b. Realizamos la llamada GET para obtener todos los datos climatológicos.
4. **SplitRegisters**
    a. Realizaremos un split de la respuesta recibida (un array de registros por día, franja horaria y factor climatológico) para crear un registro por día, franja horaria y factor climatológico.
        i. Se creará un flowfile por cada registro nuevo creado, de cara a su almacenamiento en HDFS.
5. **GetRowVariables, SetFileName, JoltAddTimeStamp**
    a. Captamos datos del registro actual, de cara a preparar el nombre de fichero.
        i. Fecha de la lectura y variable climatológica.
    b. Modificamos el nombre de fichero que asigna por defecto NIFI.
        i. [estación]_[código variable]_[fecha_lectura en formato yyyyMMdd_HHmm_HH].
    c. Mediante un procesador Jolt añadiremos una variable adicional que almacena la fecha en formato timestamp.
6. **InsertProcessedHDFS**
    a. Inserción del fichero en el sistema hdfs, en la carpeta asignada para el procesado de los datos en HIVE de la contaminación.
7. **(rama paralela) SetFileName, InsertOriginalPollutantHDFS**
    a. En esta rama se insertarán los datos originales sin procesar, ante posibles usos futuros.
    b. Se asigna un nombre de fichero (para machacar el nombre aleatorio generado por Nifi. El formato es [estación].
    c. Se inserta el fichero sin modificar en el sistema hdfs, en la carpeta asignada para el almacenamiento de datos históricos.
8. **(rama paralela) Script getDate, RouteOnContent, SetFileName, InsertIndexCityHDFS**
    a. A continuación almacenaremos la fecha procesada en el sistema HDFS. Habrá un fichero por municipio y contaminante. Almacenando la fecha más alta procesada.
    b. Mediante script groovy se recorre la respuesta, almacenando la fecha más alta para cada factor climatológico y estación.
    c. Se prepara un nombre de fichero.
        i. Un valor fijo, ya que machacamos el fichero de parámetros existente en el sistema. Formato [municipio]_[contaminante].

d. Almacenaremos en el sistema hdfs el parámetro para usos futuros.



(21.Ejemplo de registro meteorológico captado, en formato .json embellecido)

6-Entorno de Almacenamiento de datos

Si en el punto 5, hemos narrado como los datos son ingestados, como ríos que desembocan en un lago. Ahora narraremos cómo ese agua es almacenada en el lago. Trataremos como el dato será almacenado en el sistema.

6.1-Hadoop

Para ello utilizaremos Apache Hadoop (#6.1), comúnmente llamado hadoop o incluso hdfs (atendiendo al nombre utilizado por el sistema de ficheros de hadoop).

Es un proyecto que ofrece un framework, con un conjunto de aplicaciones orientadas al big data. Un sistema de ficheros distribuido, dedicado al big data de código abierto que se amolda a las necesidades de este proyecto (gran almacenamiento, bajo coste).

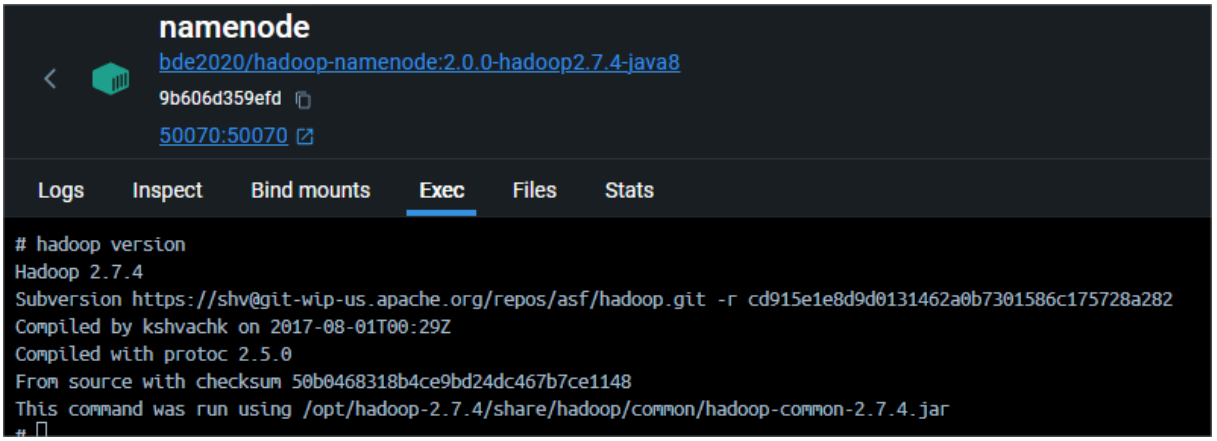
Con una arquitectura mínima requiere:

- Un nodo de nombres (namenode) que no almacena datos, sino que gestiona el acceso a los mismos
- Un nodo de datos (datanode), donde se almacenará y replicará el dato propiamente dicho.

Dentro de nuestro entorno docker hemos replicado esta configuración mínima. En la cual se alojarán los datos provenientes de Nifi. Y se post procesarán a posteriori con HIVE.

(22.Servidores en ámbito docker utilizados para el almacenamiento)

Habiendo descargado las imágenes correspondientes que corren en la versión 2.7.4



(23.Obtención de la versión de Hadoop utilizado en el proyecto)

Estos servidores fueron desplegados en el inicio a través del fichero *docker-compose.yml*, que previamente habiamos configurado, asegurando que mapea correctamente las carpetas de nuestro sistema operativo (para mantener persistencia) y que no se causaba ninguna anomalía mediante nombres o puertos incorrectos.

Podemos trabajar contra el servidor de nombres desde el apartado Exec (como se ve en la carpeta anterior lanzando comandos) o desde nuestro sistema operativo una vez lanzamos el comando: *docker exec -u 0 -it namenode bash* (para entrar con permisos de administración en el interior del nodo de nombres).

(24.Listado de las carpetas creadas en el sistema hdfs)

6.2-Estructura

La estructura de carpetas que hemos creado en el sistema hdfs se origina a partir de la carpeta TFM (en un alarde de originalidad).

- /TFM/Index
  - Almacena los índices de fecha (última fecha procesada) a nivel de contaminante y fecha meteorológica en un fichero de texto plano.
    - Será consultado desde el proceso NIFI de captación.

```
root@9b606d359efd:/# hdfs dfs -ls /TFM/Index
Found 1 items
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:53 /TFM/Index/Date
root@9b606d359efd:/# hdfs dfs -ls /TFM/Index/Date
Found 2 items
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:54 /TFM/Index/Date/City
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:56 /TFM/Index/Date/Station
root@9b606d359efd:/# hdfs dfs -ls /TFM/Index/Date/City
Found 11 items
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Manlleu_CO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Manlleu_NO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Manlleu_PM10
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Manlleu_PM2.5
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Matar?_CO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Matar?_NO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Matar?_PM10
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Vilanova i la Geltr?_CO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Vilanova i la Geltr?_NO
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Vilanova i la Geltr?_PM10
-rw-r--r--   3 nifi supergroup         10 2024-06-14 17:54 /TFM/Index/Date/City/Vilanova i la Geltr?_PM2.5
root@9b606d359efd:/# hdfs dfs -cat /TFM/Index/Date/City/Manlleu_CO
2024-06-13root@9b606d359efd:/#
```

(25.Listado de los ficheros índices creados en el sistema hdfs)

- /TFM/NifiProcessed
  - Almacena los datos ingestados desde la plataforma NIFI. Son borrados una vez procesados
- /TFM/Original
  - Almacena los datos ingestados desde la plataforma NIFI, sin ningún tipo de preprocesamiento. Se almacenan con motivo histórico, por si se quieren procesar en el futuro.
  - Están segregados por carpetas (Contaminante y Meteorología)
- /TFM/PostProcessed
  - Donde almacenamos los datos tras haberse postprocesado con Hive.
  - Almacenamos los datos tras asociar datos meteorológicos con polución y tablas externas.
  - Almacenamos todas las consultas de agregados

```
root@9b606d359efd:/# hdfs dfs -ls /TFM/PostProcessed/
Found 7 items
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/AggDay
drwxr-xr-x   - root supergroup          0 2024-06-11 21:26 /TFM/PostProcessed/AggMeteo
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/AggYear
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/Contaminants
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/Municipis
drwxr-xr-x   - root supergroup          0 2024-05-29 19:58 /TFM/PostProcessed/coreData
drwxr-xr-x   - root supergroup          0 2024-05-24 18:06 /TFM/PostProcessed/relacionMeteo
root@9b606d359efd:/#
```

(26.Listado de las carpetas donde se ubicarán los datos de las consultas agregadas en hdfs)

  - Se crea una carpeta por consulta, para mantener la coherencia de tablas externas de HIVE.

- /TFM/Predictions
    - Donde almacenamos los datos de las predicciones meteorológicas, son sobreescritos con cada ejecución.
- /TFM/Processed
    - Donde almacenamos los datos importados de Nifi para realizar una compactación de los mismos (y subsanar una incidencia asociada a la dispersión de datos por alto volumen de ficheros).
- /TFM/Sources

```
root@9b606d359efd:/# hdfs dfs -ls /TFM/Sources
Found 5 items
drwxr-xr-x   - root supergroup          0 2024-05-20 10:22 /TFM/Sources/ObjetivosCalidad
drwxr-xr-x   - root supergroup          0 2024-05-27 18:21 /TFM/Sources/historicMeteo
drwxr-xr-x   - root supergroup          0 2024-05-19 09:04 /TFM/Sources/relacionFestivos
drwxr-xr-x   - root supergroup          0 2024-05-24 18:10 /TFM/Sources/relacionMeteo
drwxr-xr-x   - root supergroup          0 2024-05-27 18:49 /TFM/Sources/relacionMeteoMetadata
```

(27.Listado de las carpetas donde se ubicarán los datos fuentes en hdfs)

    - Donde almacenamos ficheros de referencia para el postprocesado en HIVE
        - ObjetivosCalidad
            - Almacena un fichero (en formato .csv tabular) donde guardamos la relación de cada contaminante con su límite establecido (diario o anual) y si es establecido por la OMS o el BOE.
        - RelacionFestivos
            - Carpeta donde cargamos (formato .csv) un seguido de archivos con la relación de festivos.
                - Autonómicos y/o nacionales.
                - Se deposita un fichero por año, para facilitar la configuración simplemente añadiendo ficheros nuevos cada año, sin necesidad de modificar los anteriores.
        - relacionMeteo
            - Almacena un fichero .csv tabular.
            - Donde asociamos la estación meteorológica con su código, municipio asociada y una descripción
        - relacionMeteoMetadata
            - Almacena un fichero .csv tabular.
            - Donde asociamos la variable meteorológica (un entero) con su metadato (descripción).
        - historicMeteo
            - Almacenamos los datos históricos que hemos cargado en fichero.csv directamente.
    - El almacenar los ficheros dentro del sistema hdfs nos permite utilizarlos a posteriori en las consultas HIVE como si de tablas se tratara.
    - Facilita a su vez la configuración.
        - Si el usuario quiere añadir más variables al procesamiento solo debe modificar el fichero en el interior o reemplazarlo.
        - Si el usuario quiere añadir más días festivos a la lista, por ejemplo, solo deberá cargar más ficheros para ello. Facilitando su tarea.

7-Entorno de Procesamiento de dato

Hemos obtenido los datos y estos están almacenados en el sistema gracias a los pasos anteriores, ahora debemos procesarlos para obtener algo que pueda ser utilizable para obtener información a partir del dato.

Hemos realizado un preprocesamiento en NIFI, pero este era para acomodar el dato y ponernoslo más fácil en fases posteriores. Lo que buscamos ahora es transformarlo, agregarlo y modificarlo para poder permitir al usuario visualizarlo con facilidad.
Siendo todos los pasos importantes en el proceso, este es el más importante. De nuestra destreza (ayudados por el software elegido) dejaremos consultas preparadas para ser explotadas a posteriori en una herramienta de visualización.

Este "dejar consultas preparadas" no es trivial. Estamos manejando grandes, grandísimos volúmenes de datos, que pueden generar grandes tiempos de respuesta. Es gracias al trabajo en esta fase que no traspasamos este tiempo invertido a la siguiente fase, incurriendo en retrasos que sufrirá el usuario.
Se deben crear en este punto consultas, como hemos mencionado, adelantándose a las necesidades del usuario final en la medida de lo posible.No preocupandonos del tiempo invertido (eso sí, tratando de minimizarlo) en el procesado, ya que aún estamos en background de cara al resultado final.

7.1- HIVE

Para ello hemos decidido decantarnos por la herramienta HIVE (#7.1). Un sistema de data warehouse distribuido, tolerante a fallos que nos permite hacer análisis sobre grandes volúmenes de datos.

Siendo una de sus ventajas el hecho de poder ser gestionado como si fuera SQL, mediante un lenguaje muy similar (HiveQL). A su vez dispone de un conector ODBC que permite que podamos consultar el dato fácilmente de forma externa. Da un soporte completo al protocolo ACID.
Como última ventaja reseñable, podemos automatizar las consultas mediante HUE fácilmente (lo describiremos al final de este punto), dejando todo el proceso orquestado de forma automática.
HIVE tratará todo el volumen de ficheros que hemos creado como tablas. De ahí que dejásemos preparada la estructura en registros.

El contenido de una carpeta es tratado como registros de una tabla, abstrayéndose de los ficheros en sí y atacando el dato contenido (siempre que se comparta una estructura).

Para ello hemos optado por trabajar con "tablas externas". HIVE ofrece la posibilidad de crear este sistema, en el cual solo definimos el metadato de la tabla (columnas, formato, ubicación) permitiendo al sistema trabajar con el dato directamente. A su vez nos ofrece un añadido de seguridad. Si borramos la tabla, en realidad solo estamos borrando el metadato, no el dato en sí, que permanecerá inalterado.

Hemos de reconocer que se han tenido que borrar muchas veces las tablas durante el desarrollo inicial. Para modificarlas, agregar columnas, cambiar la ubicación del dato… Esto ha sido mucho más

fácil y seguro, habiéndonos desapegado del dato en sí y trabajando contra el metadato, gracias al hecho de ser tablas externas.

```
#creamos la tabla de final de meteo
CREATE EXTERNAL TABLE IF NOT EXISTS processed_meteo (id string,codi_estacio string,codi_variable string,data_lectura string,valor_lectura string,codi_base string,timeStampSecond string)
COMMENT 'storage of processed meteorological data'
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS TEXTFILE
LOCATION '/TFM/Processed/Meteo/';
```

```
hive> describe processed_meteo
    > ;
OK
id                      string                  from deserializer
codi_estacio            string                  from deserializer
codi_variable           string                  from deserializer
data_lectura            string                  from deserializer
valor_lectura           string                  from deserializer
codi_base               string                  from deserializer
timestampsecond         string                  from deserializer
Time taken: 1.515 seconds, Fetched: 7 row(s)
hive>
```

(28.Ejemplo de script de creación de tabla en HIVE y estructura de tabla resultante en HIVE)

Se incluyen ejemplos de creación de las tablas en los anexos. Para el registro completo, se debe consultar la entrega de código realizada durante el proyecto, donde se incluyen un fichero.txt con todos los comandos para la creación de tablas.

Para poder trabajar con el sistema HIVE hemos desplegado en nuestro sistema docker varios servidores (hive-metastore, hive-server, database,hive-metastore-postgresql y hue).

(29.Listado de servidores utilizados para la utilización de HIVE en nuestro entorno)

7.2-Estructura de las consultas

Dentro del sistema HIVE, las consultas son similares a las que pudieran ser realizadas en un sistema SQL, pudiendo hacer SELECTs, INSERTs, UPDATEs sin problemas. Utilizando los ficheros que hemos generado en la ingesta como fuente de datos (o los ficheros fuentes que usamos de referencia), donde cada línea de esos ficheros es un registro.

Estructuramos las consultas realizadas en:

- **Consultas de compactación.**

    - Los ficheros provenientes de NIFI son insertados en una carpeta de datos procesado. Para pasar de una miríada de pequeños ficheros a un pequeño número de ficheros más grandes.
    - Es necesario para poder procesar las consultas de relación en tiempos aceptables. Sino la máxima dispersión en el sistema Hadoop de ficheros ralentiza la consulta.
- **Consultas de limpieza.**
    - Donde eliminamos los registros de la carpeta donde se han ingestado los datos de NIFI y evitar duplicidades.
    - El sistema no nos permite hacer un DELETE de registros en las condiciones que tenemos configuradas, con lo que simplemente realizamos un INSERT OVERWRITE (que machaca los datos existentes) con un filtro que no genera resultado (a la práctica borrando el contenido de la carpeta).
    - También incluimos la limpieza en tablas procesadas donde pueda haber algún dato duplicado o que se haya quedado descolgado.
- **Consulta de relación.**
    - La consulta que genera el core de nuestro dato relacionando:
        - Los datos ingestados de contaminación
        - Los datos ingestados de meteorología (relacionando por fecha y hora)
        - Nuestra tabla de días festivos, para identificar si la fecha procesada corresponde a un día festivo (incluimos en este cálculo el domingo como día festivo)
        - La tabla de metadatos meteorológicos (para tener la descripción de la variable meteorológica que nos interesa)
        - Establecemos la relación entre estaciones de polución y meteorología a través de nuestra tabla intermedia de relación de estaciones.
    - Esta consulta aumenta el número de registros. Ya que para cada municipio contaminante y fecha se genera un registro por cada variable meteorológica.
    - Esto se enfoca de cara a poder realizar debidamente los análisis posteriores.
- **Consultas de agregado.**
    - Debido al volumen de registros (y el que se prevé en el futuro) hemos establecido una serie de consultas para realizar los agregados de los resultados obtenidos.
    - Mostramos el máximo de datos posible que puedan ser de utilidad para las consultas posteriores (municipio, nombre de estación, coordenadas de geoposicionamiento, el contaminante y condición meteorológica medida).
    - De mayor a menor número de registros
        - Agregado por hora
        - Agregado por día
        - Agregado por año
    - Obviamente en función de las consultas que se quieran realizar a posteriori se debe escoger entre una fuente u otra. Siendo la que más detalle puede aportar la agregada por hora.
- **Consultas de muestreo**
    - Donde almacenamos los valores únicos de determinados valores para facilitar filtros posteriores
        - Contaminantes

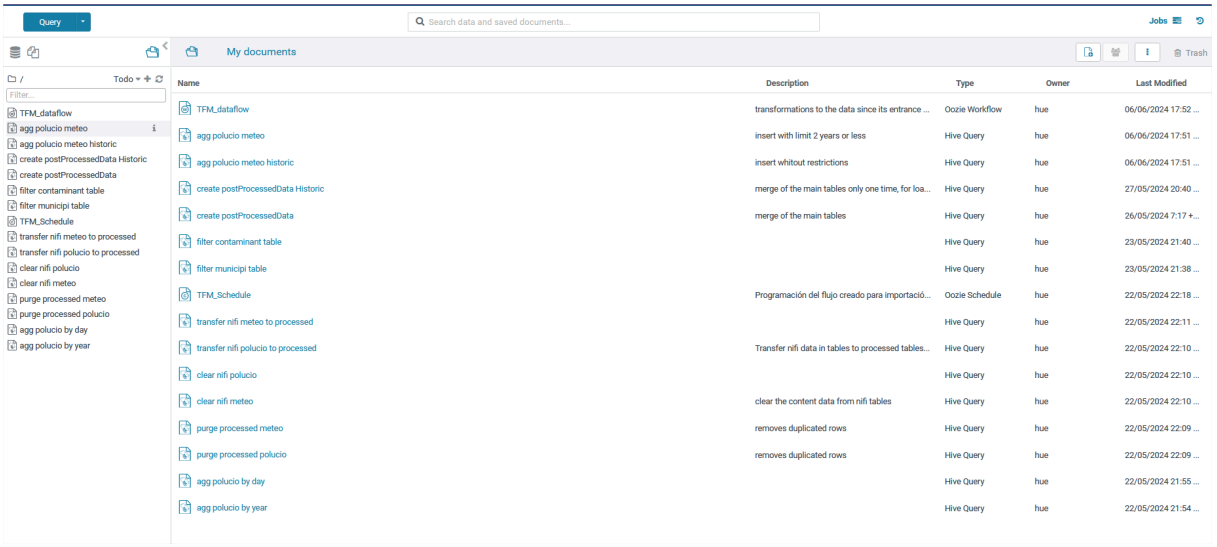
- Municipios

Se debe tener en cuenta que todas estas consultas dejan su resultado almacenado en el sistema (machacando el anterior en su procesamiento) de tal forma que no se recalcula cuando es solicitado desde fuentes externas. Es aportado directamente, eliminando el procesamiento (que ha sido realizado previamente).

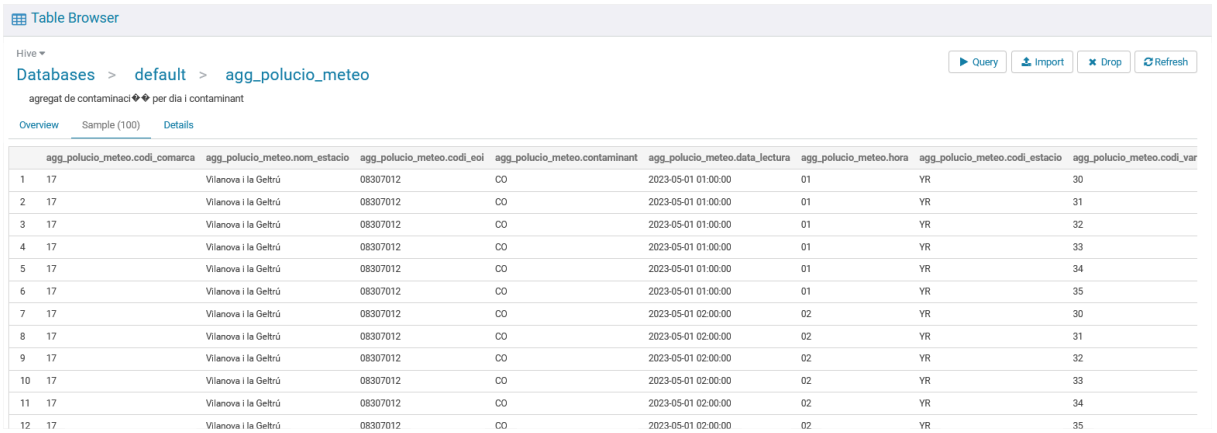
7.3- Programación de las consultas

De cara a programar las consultas utilizamos Apache Hue (#7.2). Un asistente para trabajar con bases de datos y data warehouses.
Es fácil integrarlo contra el sistema Hadoop. Simplemente acoplando un servidor Hue y MySql a nuestro sistema y configurandolo para que se conecten a nuestro sistema Hadoop.

Nos permite almacenar las consultas en el Mysql de forma que podemos lanzarlas mediante una interfaz gráfica, mostrando el resultado. Y nos permite acceder a nuestras bbdds en Hive de forma ágil y sencilla con su interfaz gráfica.



(30.Muestra de las consultas y procesos de orquestación creados en la interfaz HUE)



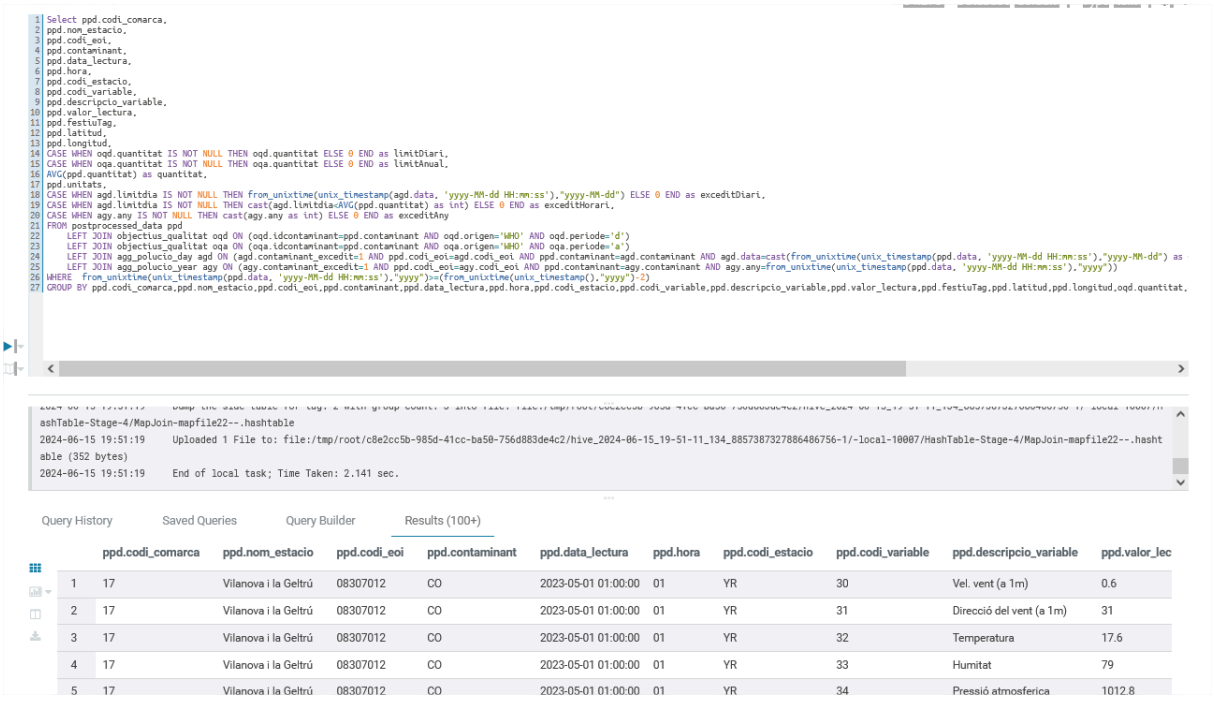
	agg_polucio_meteo.codi_comarca	agg_polucio_meteo.nom_estacio	agg_polucio_meteo.codi_eoi	agg_polucio_meteo.contaminant	agg_polucio_meteo.data_lectura	agg_polucio_meteo.hora	agg_polucio_meteo.codi_estacio	agg_polucio_meteo.codi_var
1	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	30
2	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	31
3	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	32
4	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	33
5	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	34
6	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	35
7	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	30
8	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	31
9	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	32
10	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	33
11	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	34
12	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 02:00:00	02	YR	35

(31.Ejemplo de consulta a una tabla en HUE)

Dentro del sistema Hue hemos almacenado todas las consultas mencionadas en el punto anterior (que pueden ser importadas y exportadas como ficheros .json).

- Consultas de compactación
    - transfer nifi meteo to processed
    - transfer nifi polución to processed
- Consultas de limpieza
    - clear nifi polucio
    - clear nifi meteo
    - purge processed meteo
    - purge processed polucio
- Consultas de relación
    - create postProcessedData Historic
    - create postProcessedData
        - relaciona los últimos dos años de datos
- Consultas de agregado
    - agg polucio meteo
        - Muestrea únicamente por hora los dos últimos años
    - agg polución meteo historic
        - Sin limite de fecha
    - agg polucio by day
    - agg polucio by year
- Consultas de muestreo
    - filter contaminant table
    - filter municipi table

Una ventaja añadida del sistema Hue es que nos facilita mucho el poder hacer consultas para verificar los datos y nos ofrece un histórico de las consultas realizadas.
Permitiéndonos incluso saber el tiempo que ha tardado en realizarse una consulta.



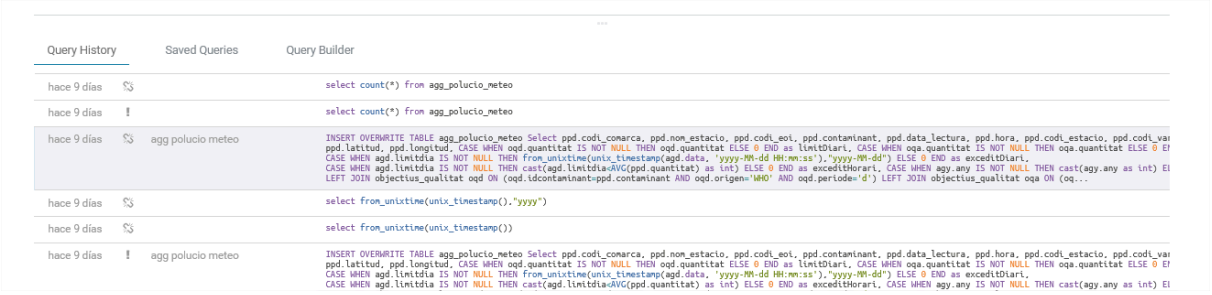
```
Select ppd.codi_comarca,
ppd.nom_estacio,
ppd.codi_eoi,
ppd.contaminant,
ppd.data_lectura,
ppd.hora,
ppd.codi_estacio,
ppd.codi_variable,
ppd.descripcio_variable,
ppd.valor_lectura,
ppd.festiuTag,
ppd.latitud,
ppd.longitud,
CASE WHEN oqd.quantitat IS NOT NULL THEN oqd.quantitat ELSE 0 END as limitDiari,
CASE WHEN oqa.quantitat IS NOT NULL THEN oqa.quantitat ELSE 0 END as limitAnual,
AVG(ppd.quantitat) as quantitat,
ppd.unitats,
CASE WHEN agd.limitdia IS NOT NULL THEN from_unixtime(unix_timestamp(agd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") ELSE 0 END as exceditDiari,
CASE WHEN agd.limitdia IS NOT NULL THEN cast(agd.limitdia<AVG(ppd.quantitat) as int) ELSE 0 END as exceditHorari,
CASE WHEN agy.any IS NOT NULL THEN cast(agy.any as int) ELSE 0 END as exceditAny
FROM postprocessed_data ppd
    LEFT JOIN objectius_qualitat oqd ON (oqd.idcontaminant=ppd.contaminant AND oqd.origen='WHO' AND oqd.periode='d')
    LEFT JOIN objectius_qualitat oqa ON (oqa.idcontaminant=ppd.contaminant AND oqa.origen='WHO' AND oqa.periode='a')
    LEFT JOIN agg_polucio_day agd ON (agd.contaminant_excedit=1 AND ppd.codi_eoi=agd.codi_eoi AND ppd.contaminant=agd.contaminant AND agd.data=cast(from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") as
    LEFT JOIN agg_polucio_year agy ON (agy.contaminant_excedit=1 AND ppd.codi_eoi=agy.codi_eoi AND ppd.contaminant=agy.contaminant AND agy.any=from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy"))
WHERE  from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy")>=(from_unixtime(unix_timestamp(),"yyyy")-2)
GROUP BY ppd.codi_comarca,ppd.nom_estacio,ppd.codi_eoi,ppd.contaminant,ppd.data_lectura,ppd.hora,ppd.codi_estacio,ppd.codi_variable,ppd.descripcio_variable,ppd.valor_lectura,ppd.festiuTag,ppd.latitud,ppd.longitud,oqd.quantitat,
```

```
ashTable-Stage-4/MapJoin-mapfile22--.hashtable
2024-06-15 19:51:19     Uploaded 1 File to: file:/tmp/root/c8e2cc5b-985d-41cc-ba50-756d883de4c2/hive_2024-06-15_19-51-11_134_8857387327886486756-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile22--.hasht
able (352 bytes)
2024-06-15 19:51:19     End of local task; Time Taken: 2.141 sec.
```

Query History | Saved Queries | Query Builder | Results (100+)

	ppd.codi_comarca	ppd.nom_estacio	ppd.codi_eoi	ppd.contaminant	ppd.data_lectura	ppd.hora	ppd.codi_estacio	ppd.codi_variable	ppd.descripcio_variable	ppd.valor_lec
1	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	30	Vel. vent (a 1m)	0.6
2	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	31	Direcció del vent (a 1m)	31
3	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	32	Temperatura	17.6
4	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	33	Humitat	79
5	17	Vilanova i la Geltrú	08307012	CO	2023-05-01 01:00:00	01	YR	34	Pressió atmosferica	1012.8

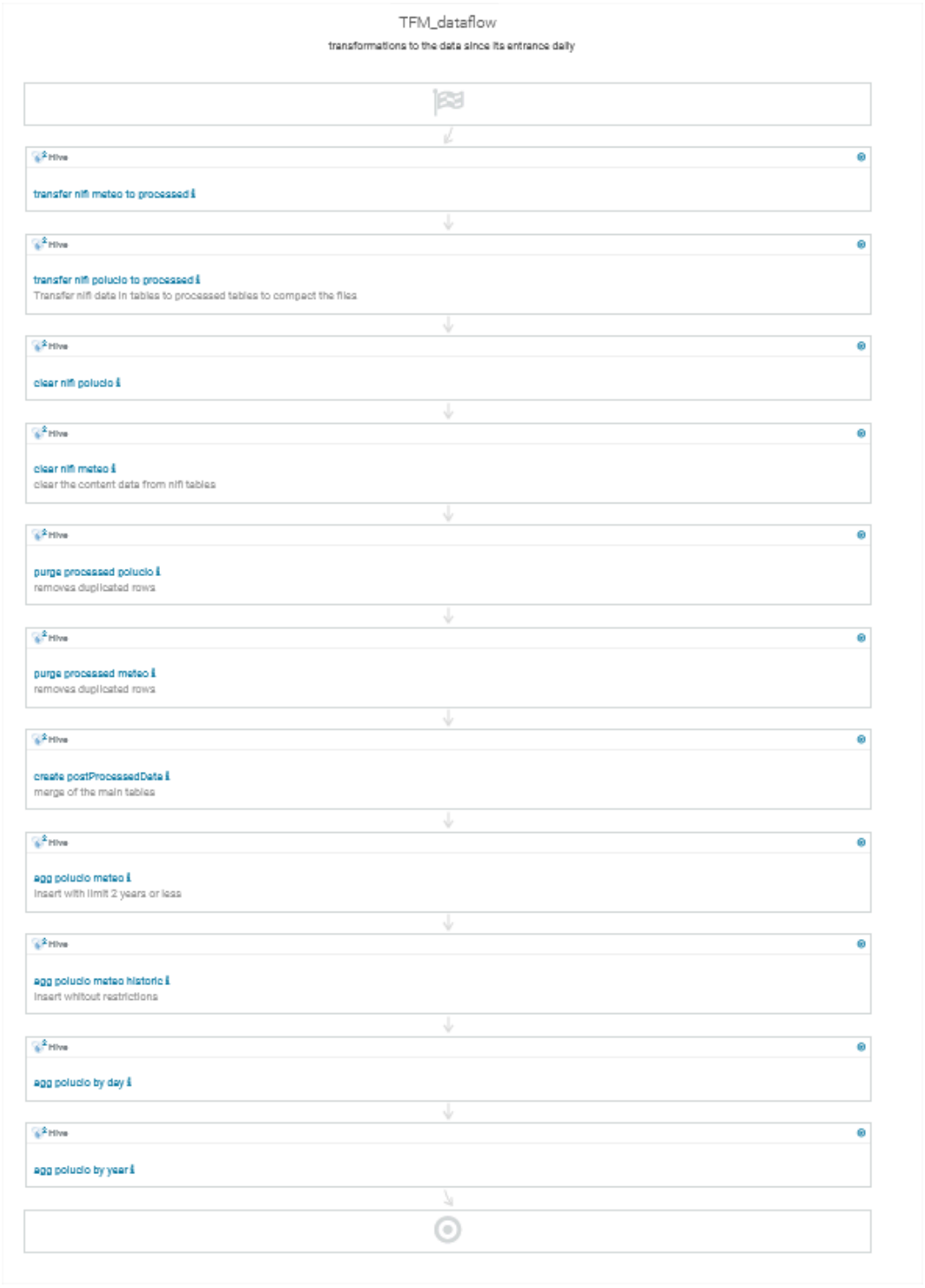
(32.Ejemplo de resultado de query lanzada en entorno HUE)



(33.Muestra de histórico de consultas en entorno HUE)

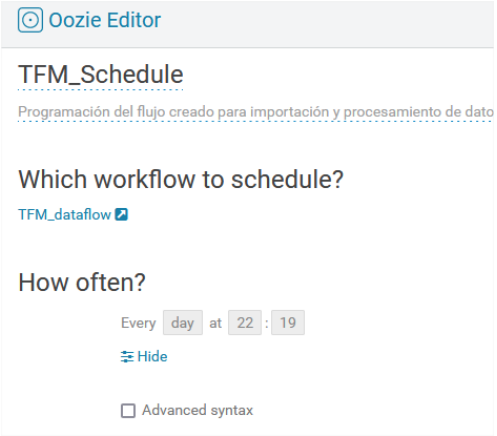
Por último, nos permite establecer programación en las consultas.
Primero establecemos un flujo de trabajo, donde estableceremos un seguido de consultas y qué hacer ante la ejecución exitosa (o no) de una consulta.

- En nuestro caso de trabajo estableceremos en orden todas nuestras consultas para nutrir y recrear nuestros datos a externalizar.



(35.Flujo de procesos establecido en HUE para ejecución orquestada)

Una vez creado un flujo de trabajo, le asignamos una programación para que se ejecute de forma desatendida. Ejecutándose trás la captación de los datos para ejecutarse de forma óptima.



(36.Ejemplo de programación de proceso)

8-Entorno de salida de dato - Power Bi

Hemos captado el dato, lo hemos procesado y hemos jugado con él. Ahora entramos en la fase final, donde visualizamos el dato al usuario.

Gracias al trabajo realizado en puntos anteriores, tenemos una masa de datos procesados que pueden ser explotados fácilmente. Pudiendo ser visualizado con cualquier herramienta de visualización del mercado sin problema (Tableau, Powerbi, GoogleCharts, Zoho…), solo necesitando conectarla a nuestra fuente de datos, no habiendo desarrollado específicamente para ninguna para facilitar cambios tecnológicos futuros.

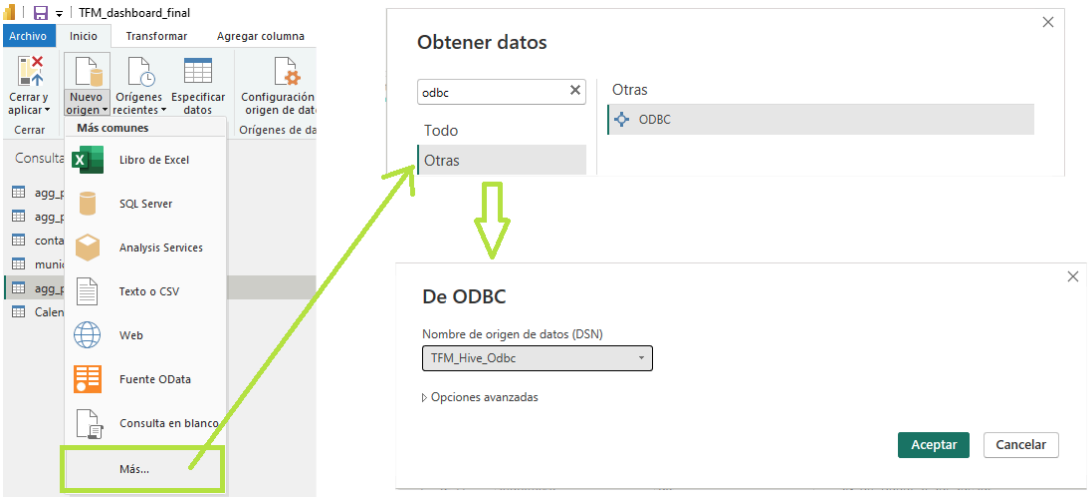
En este proyecto hemos decidido utilizar PowerBi, una herramienta de Microsoft que nos ofrecía un período de pruebas adecuado para mostrar el resultado del mismo. Una plataforma de Business Intelligence que nos permite conectar contra nuestro origen de datos Hadoop mediante una conexión ODBC.

Dentro de Power Bi se deben seguir unos sencillos pasos para su utilización.

8.1 Captación del dato y formateo

La captación del dato es un punto sencillo. Solo debemos informar el origen de datos a la herramienta Power BI.

Desde la sección de Modelado. Seleccionamos “Nuevo Origen”->Más->Odbc-> [Seleccionamos el origen odbc que previamente hemos configurado contra nuestras tablas HIVE]



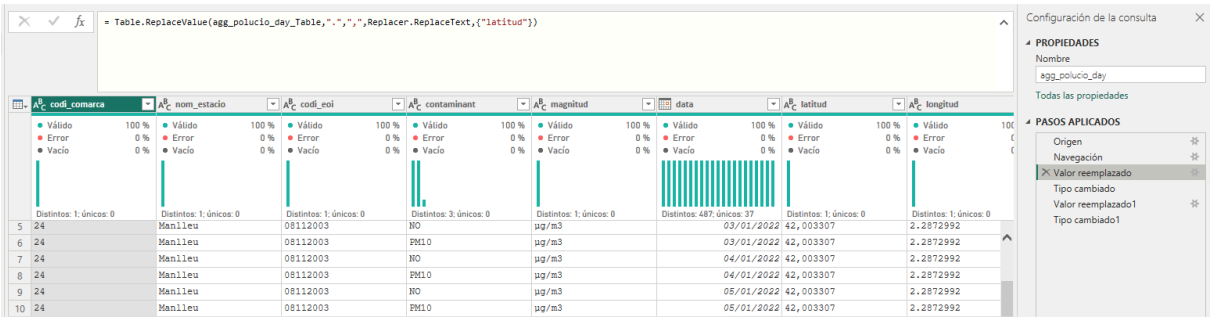
(37.Obtención de origen de datos ODBC)

Una vez establecido el origen de datos seleccionaremos las tablas con las que queremos trabajar (definidas en los puntos anteriores). En nuestro caso:

- agg_polucio_meteo
- agg_polucio_day
- agg_polucio_year
- contaminants
- municipis

Una vez cargadas, formateamos las tablas. Como el dato viene ya muy preformateado los cambios son mínimos para cada tabla.

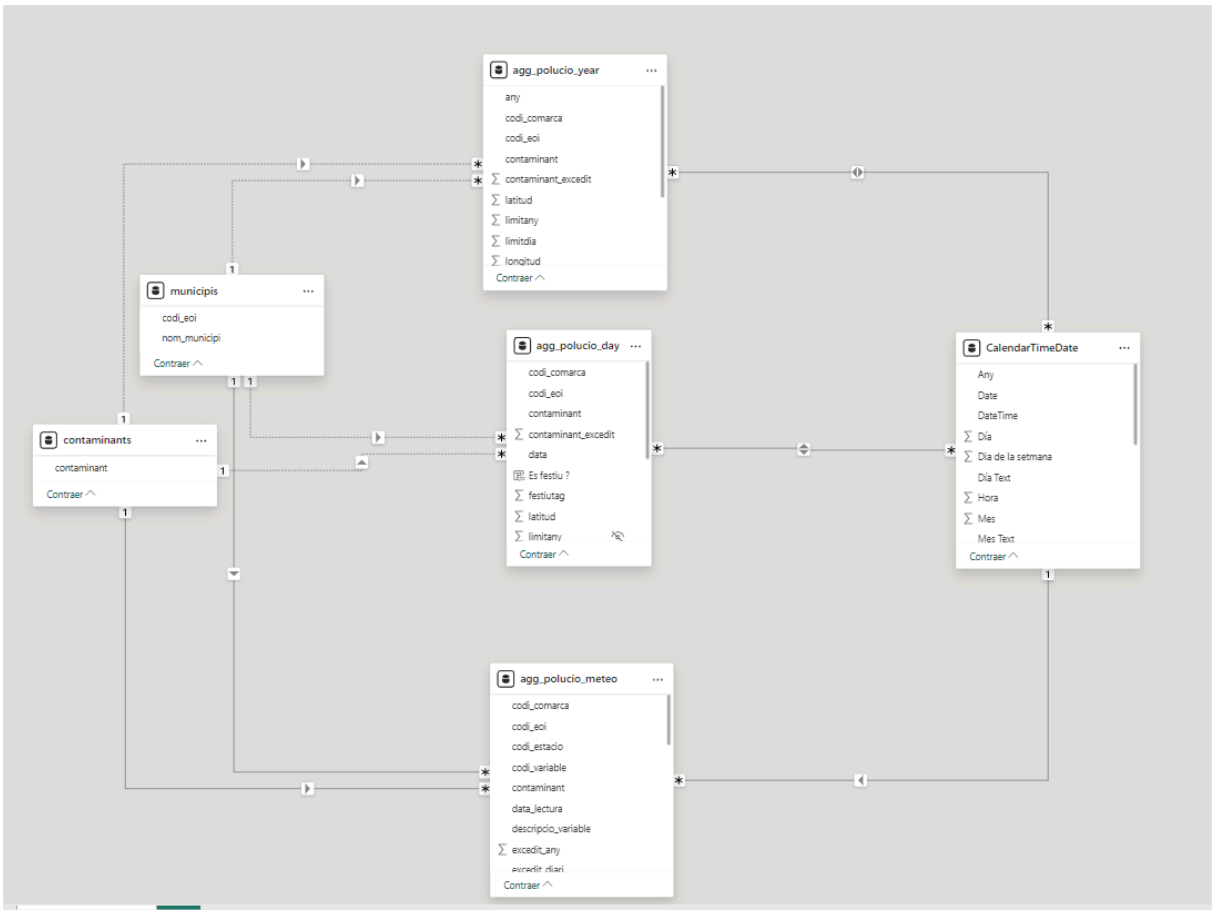
- Cambios de tipo de dato
    - De tipo string a date en las columnas fecha, para trabajar adecuadamente con esas columnas.
    - De tipo string a int (no se reconoció el tipo de dato que constaba correctamente en HIVE en estos casos).
    - Reemplazo de . por , en las columnas latitud y longitud. Para adecuarlo con el componente que lo procesa en Power Bi para el geoposicionamiento.



(38.Ejemplo de formateo de tabla y operaciones realizadas)

8.2-Elaboración de modelo

Una vez establecido el origen de datos y las tablas con las que trabajamos debemos crear el modelo que nos permitirá posteriormente visualizar el dato. Para ello estableceremos las relaciones entre las diferentes tablas en la sección de modelado y relación.



(39.Modelo de tablas utilizada)

Una vez se inició la elaboración de los informes se percibió de necesidades en el modelo que no se detectaron en el momento inicial. Pudiendo subsanarse fácilmente con modificaciones posteriores.

- De cara a mejorar los filtrados y que un informe pudiese afectar (una vez aplicados los filtros) al resto es necesario utilizar tablas intermedias con:
    - Contaminantes
    - Municipios

  Estas tablas están relacionadas con las tablas de agregados para facilitar el filtrado a través múltiples informes.
- Con el mismo espíritu se ha elaborado una tabla de fechas, con componente fecha y hora (con saltos cada hora, adecuado a las lecturas de los sensores)

Siendo la tabla de fechas elaborada mediante una fórmula DAX (incorporada en los anexos) y las tablas de contaminantes y municipios elaboradas rápidamente en HIVE (realizando y almacenando consultas con la clave DISTINCT).

Una vez creadas estas tablas intermedias en nuestro modelo establecemos las relaciones pertinentes con la cardinalidad necesaria entre ellas. Siendo lo normal en relación 1:N y estableciendo una N:N en el caso de las tablas de agregado de día y año por la naturaleza del campo fecha que almacenan.

8.3 Elaboración de los informes.

Una vez tenemos los datos linkeados en PowerBi y el modelo realizado, solo tenemos que establecer los informes para poder explotar el dato convenientemente.
Para ello incluiremos los gráficos, filtros y análisis necesarios para poder ser de utilidad para el usuario.
Procederemos a describir los 4 informes realizados.

8.3.1-Informe Evolución de contaminantes

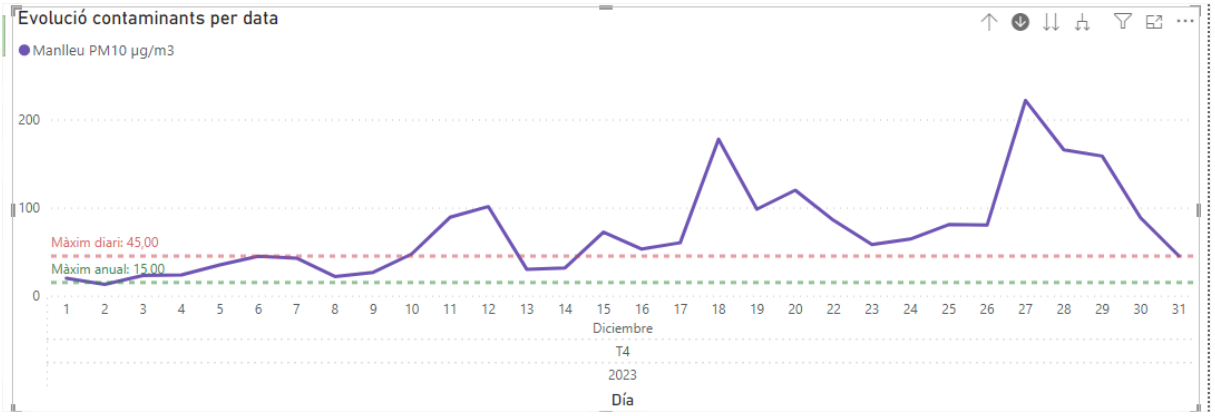
(40.Informe de evolución de la contaminación. Para Manlleu y NO)

Este informe permite ver la evolución de los contaminantes estudiados para los municipios seleccionados.
En el gráfico superior se ve la evolución del contaminante para los municipios seleccionados (en este caso PM10 para Manlleu y Mataró).

- Mostramos las líneas límites de contaminación diaria y anual como referencia.
- Mostramos la línea de progreso (punteada), para reflejar la evolución.

En el gráfico inferior se utiliza el componente predictivo de Power Bi para ver la evolución a tiempo futuro (el área gris en la sección derecha del gráfico).

Ambos gráficos permiten navegar por la jerarquía establecida de fecha, para hacer un análisis más fino.



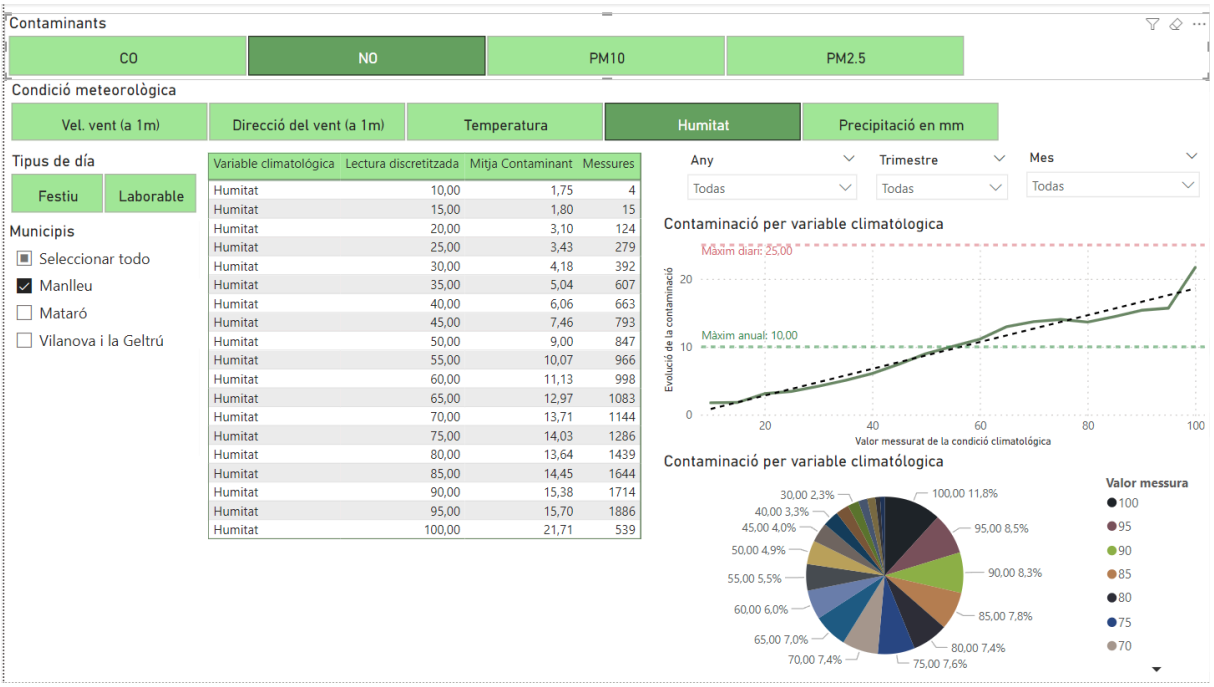
(41.Estudio de la evolución de la contaminación adentrándose en jerarquía de fechas)

Se permite filtrar por:

- Contaminante (CO,NO,PM2.5,PM10) (multiselección)
- Municipio (multiselección)
- Discriminación entre solo evolución considerando días festivos, laborables o ambos.

En la sección inferior izquierda se muestran las tarjetas indicativas que nos permiten saber los episodios anuales, diarios u horarios donde se ha superado el límite (durante el periodo de fechas analizado)

8.3.2-Informe Contaminants por metereología



(42.Informe de evolución de contaminación según el clima)

En este informe se mide la evolución de la contaminación y la influencia en esta según las condiciones meteorológicas.

Nos permite cruzar las mediciones meteorológicas y ver si ante una variación de una condición meteorológica se espera un aumento o descenso de contaminación futuro

Nos permite filtrar por:

- Contaminante (CO,NO,PM2.5,PM10) (multiselección)
- Variable climatológica (velocidad del viento, dirección del viento, humedad, presión, temperatura).
- Municipio (multiselección)
- Discriminación entre solo la evolución considerando días festivos, laborables o ambos.
- Filtro desplegable por año, trimestre, mes para establecer periodos más acotados de análisis.

Este informe nos es muy útil de cara a la predicción, poder saber qué esperar ante una condición climatológica futura.

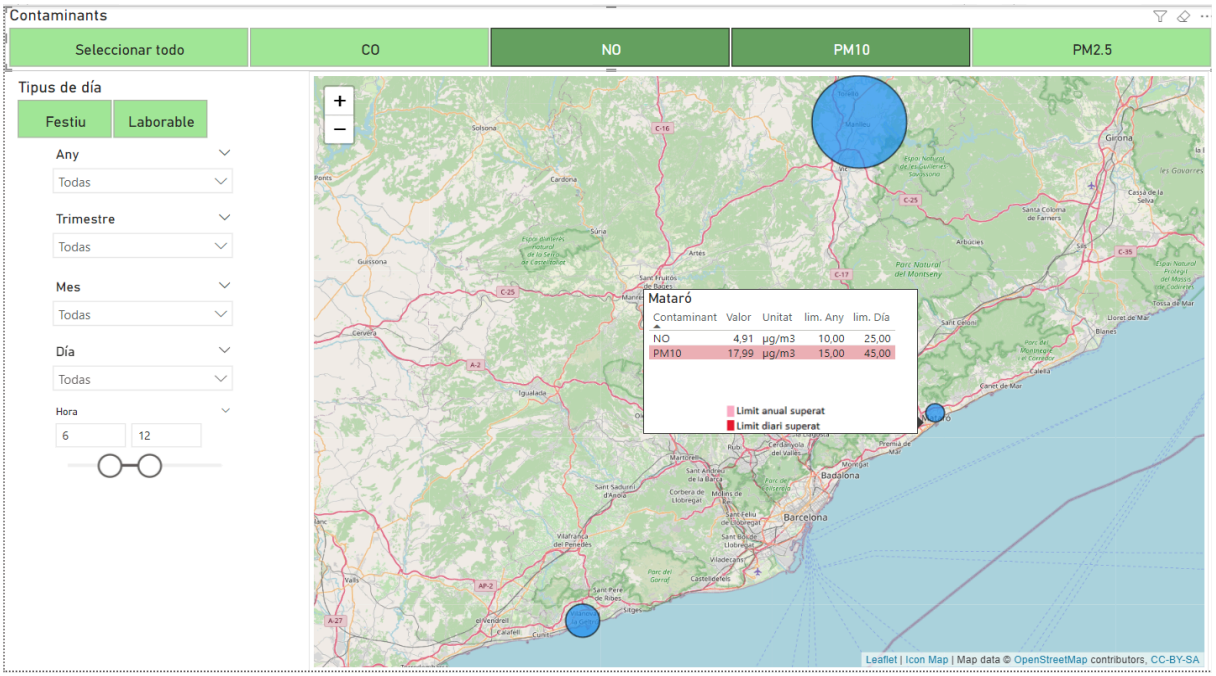
De paso nos ha dado interesantes datos sobre saber que condiciones meteorológicas afectan o no a la contaminación.
En la parte central izquierda nos ofrece una tabla con datos para saber la media de contaminación por la variable climatológica escogida (hemos discretizado los datos) y su peso en forma de número de mediciones que corresponden.

En la parte central derecha hemos mostrado esa tabla en formato de gráfico para reflejar la evolución.

- Mostramos las líneas límites de contaminación diaria y anual como referencia.
- Mostramos la línea de progreso (punteada), para reflejar la evolución.

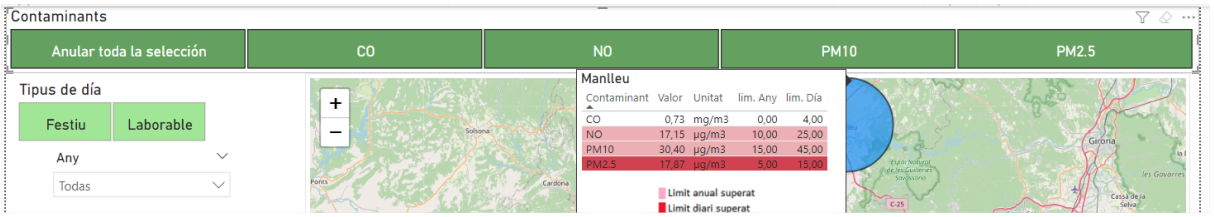
Siendo la parte inferior para mostrar el peso de cada rango meteorológico respecto a la medición de contaminación.

8.3.3-Informe Geo posicionamiento



(43.Informe de contaminación según posicionamiento)

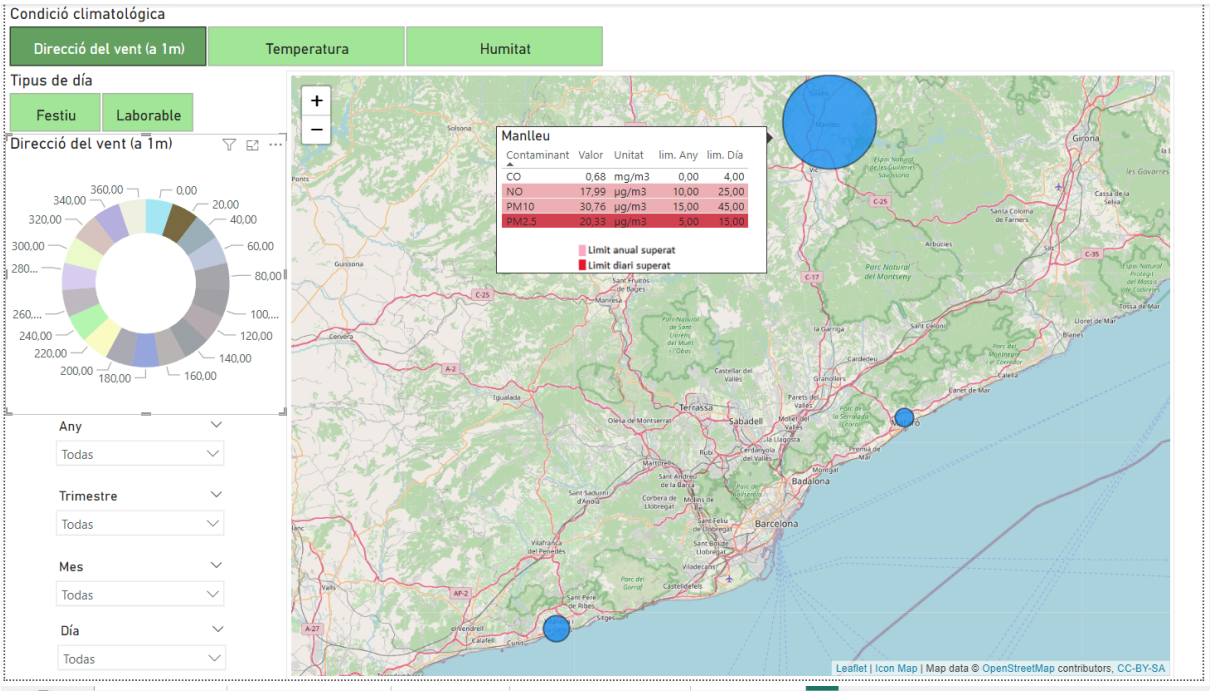
Este informe nos permite mostrar en un mapa por cada municipio evaluado la evolución del contaminante analizado.
Mostrando un círculo de tamaño variable según la importancia de la contaminación medida. Mayor, cuanta más contaminación presente.
Si seleccionamos un municipio nos muestra una tarjeta que identifica el municipio, la medición obtenida e indica con color si se ha superado o no los límites diarios y anuales.



(44. Ejemplo de visualización de tarjeta de datos por municipio)
Este informe nos permite filtrar por:

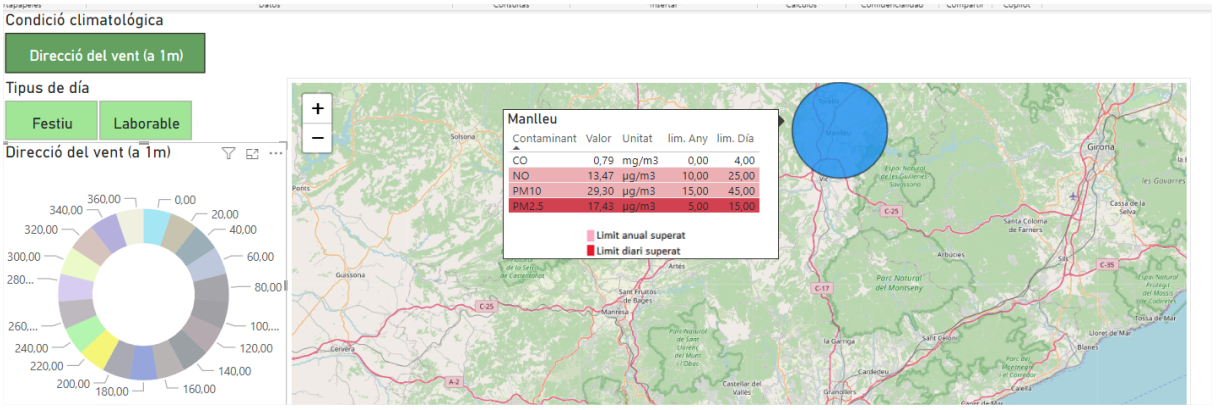
- Contaminante (multivalor)
- Rango temporal
    - No se ha realizado el filtro de forma jerarquizada para poder tener una visión más amplia.
        - Por ejemplo, poder filtrar todos los datos del primer trimestre o de todos los lunes registrados.
    - Incluyendo un slide con horas para poder evaluar sólo determinados rangos horarios.

8.3.4-Informe Geo posicionamiento y climatología



(45.Informe de contaminación y su evolución según la climatología en un mapa)

Este informe nos permite ver sobre un mapa la evolución de la contaminación. Mostrándonos en las tarjetas informativas por municipio como evoluciona según el factor climatológico escogido y su valor (en la captura hemos escogido dirección del viento en 20º, para ver la evolución de la contaminación en un viento de Barcelona a Manlleu)



(46.Uso de filtro según viento y resultado en la ciudad de Manlleu en su evolución de contaminación)

Seleccionando un viento opuesto se observa un descenso, pero ligero.
Con este informe la intención es explorar las variaciones climatológicas y su influencia en las múltiples ciudades evaluadas. No todas tienen que ser afectadas por igual ante el mismo factor.
Este informe nos permite filtrar por:

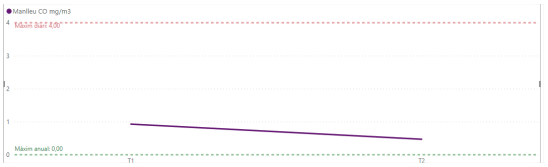
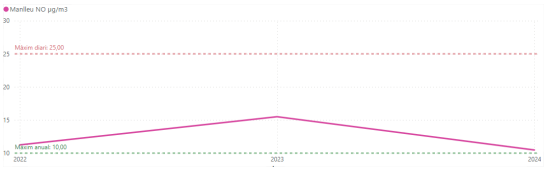
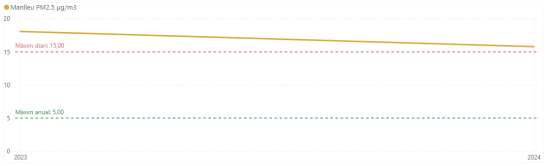
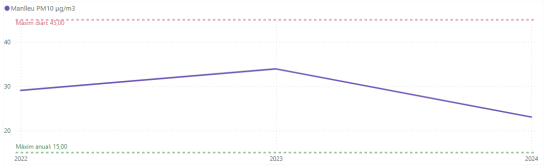
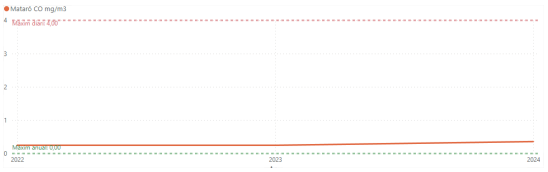
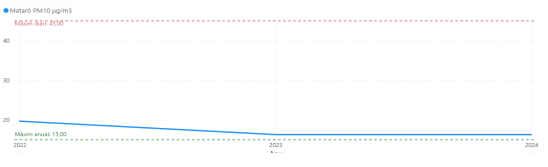
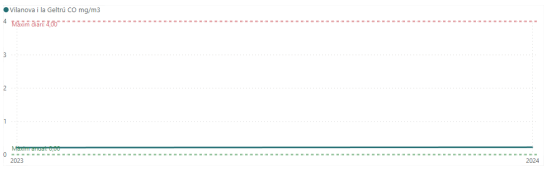
- Factor climatológico
- Rango temporal
    - No se ha realizado el filtro de forma jerarquizada para poder tener una visión más amplia.
        - Por ejemplo, poder filtrar todos los datos del primer trimestre o de todos los lunes registrados.
    - Incluyendo un slide con horas para poder evaluar sólo determinados rangos horarios.

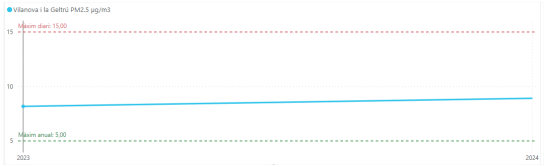
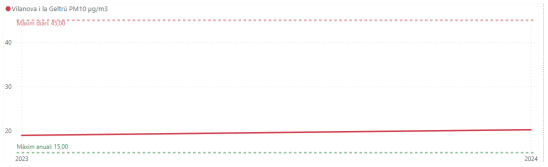
9-Resultados

Hemos realizado todo el desarrollo de un lago de datos desde el principio hasta el final. Se han analizado las fuentes, las hemos ingestado, procesado y explotado finalmente en informes.
Ahora es el momento de mostrar los resultados obtenidos en función de la contaminación y del clima.

Primero mostraremos la evolución de cada contaminante por municipio.

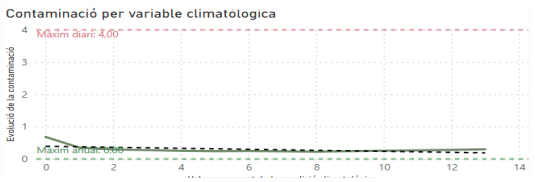
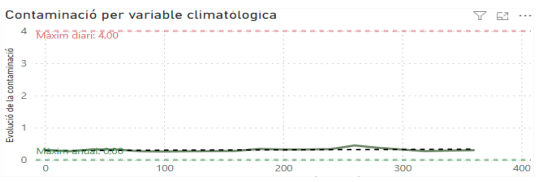
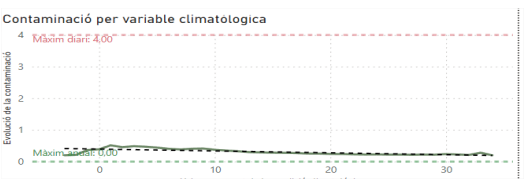
Municipio	Contaminante	Evolución	Gráfico de evolución

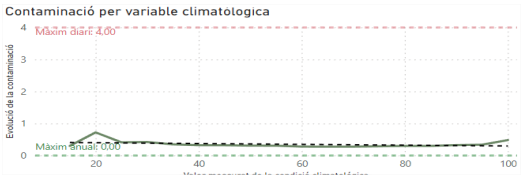
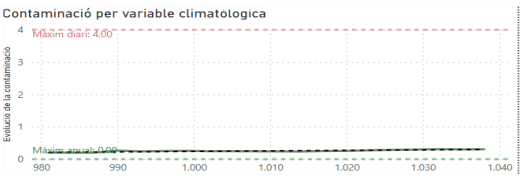
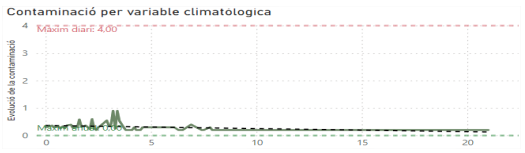
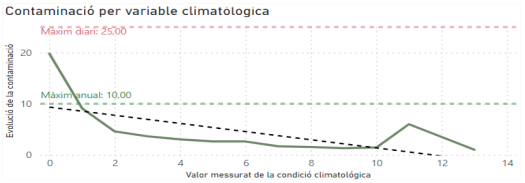
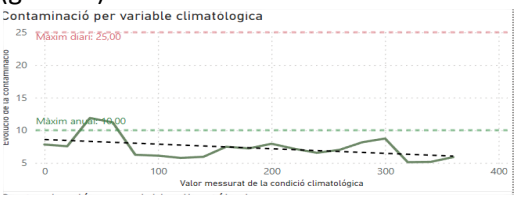
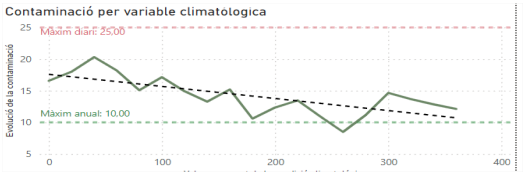
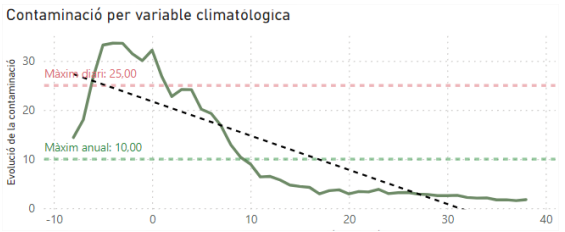
Manlleu	*CO*	Se registran valores muy bajos de CO, a partir del 2024 cuando empezó a medirse. Muy por debajo del límite diario (no hay límite anual).	
	NO	Se registra un claro descenso desde el repunte de 2023, por encima del límite anual.	
	PM2.5	Se percibe un ligero descenso en la actualidad. Por encima del límite anual y límite diario.	
	PM10	Después de un repunte en 2023 se ha registrado un claro descenso. Aún así por encima de los límites anuales.	
Mataró	*CO*	Se registran valores muy bajos de CO. Muy por debajo del límite diario (no hay límite anual).	
	NO	Se percibe un ligerísimo aumento muy por debajo de los límites de seguridad (tanto anual como diario)	
	PM2.5	No se miden las PM2.5 en los sensores de Mataró	
	PM10	Se percibe un ligero aumento en la contaminación por PM10.	
Vilanova i la Geltrú	*CO*	Se registran valores muy bajos de CO. Muy por debajo del límite diario (no hay límite anual).	

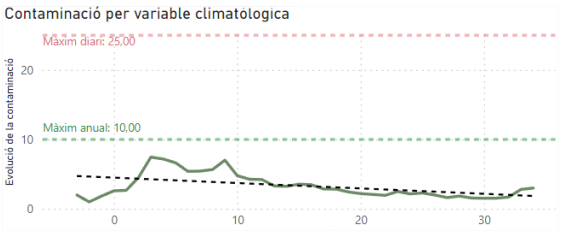
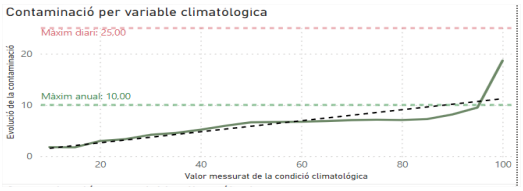
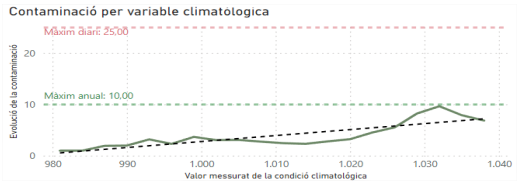
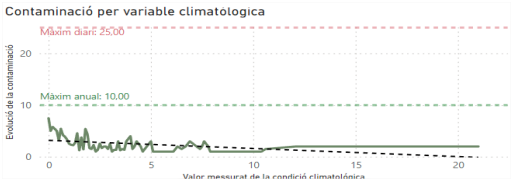
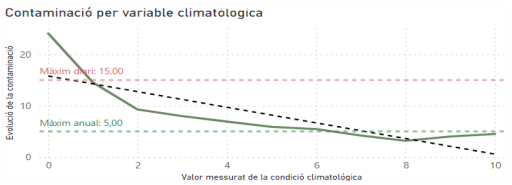
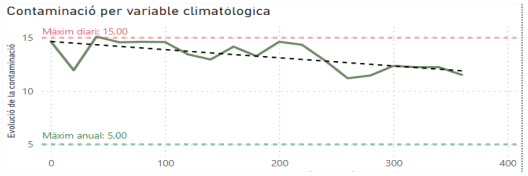
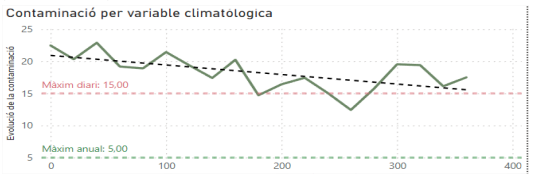
	NO	Se percibe un ligero aumento por debajo de los límites de seguridad (tanto anual como diario)	
	PM2.5	Se advierte un aumento de la contaminación por encima del límite anual, pero no el diario.	
	PM10	Se advierte un ligero aumento de la contaminación por encima del límite anual, pero no el diario.	

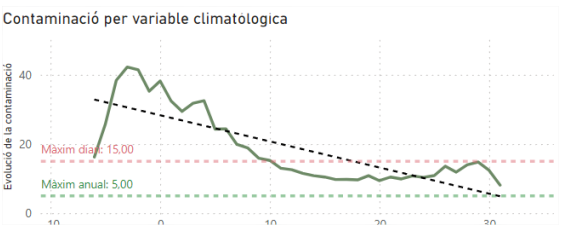
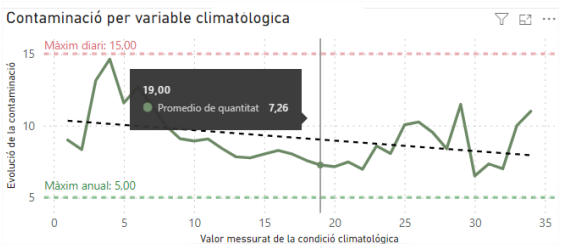
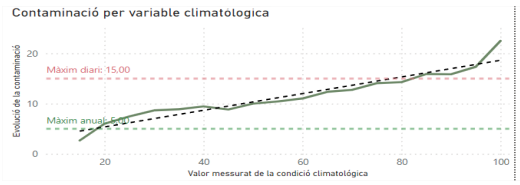
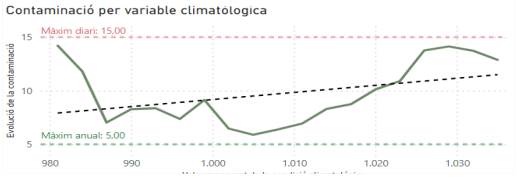
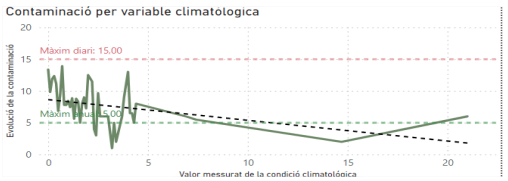
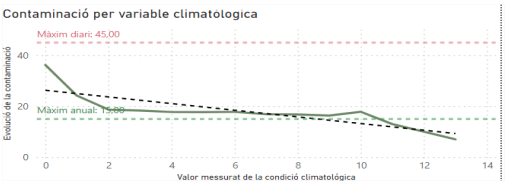
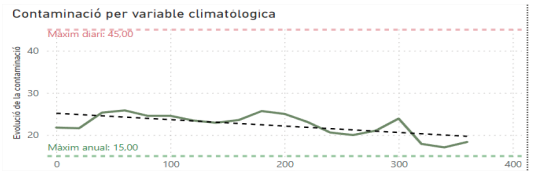
A continuación mostraremos los resultados analizados de la influencia del clima sobre la contaminación.

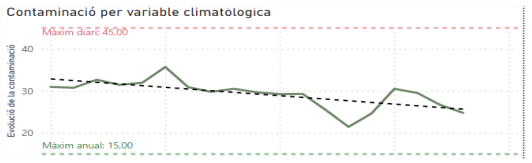
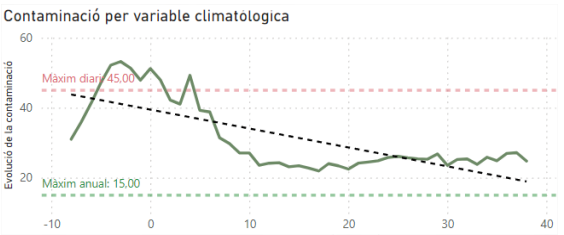
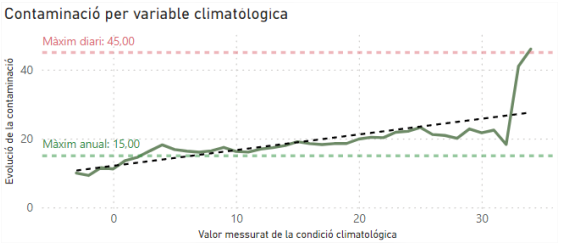
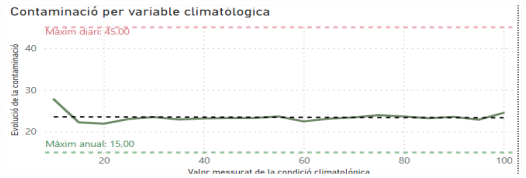
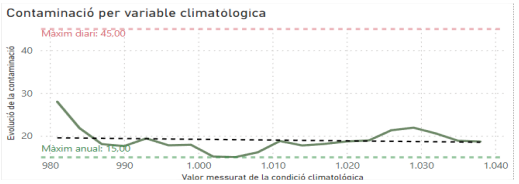
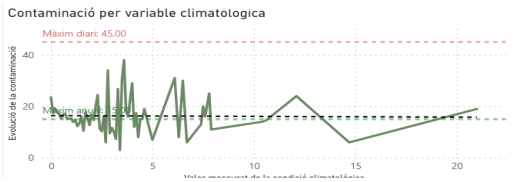
Para simplificar y no sobreextender la longitud del texto, lo hemos realizado sobre todos los municipios a la vez. Menos en el caso de dirección del viento, donde se han percibido detalles interesantes (focalizados en Manlleu). Se detectan casos de aumento claros ante vientos de componente norte-noreste.

Contaminante	Factor climatológico	Evolución	Gráfico de evolución
CO	*Vel. Viento (1m)*	No se percibe una modificación susceptible	
	Dir. Viento (1m)	No se percibe una modificación susceptible de forma general	
	Temperatura	No se percibe una modificación susceptible de forma general	

	Humedad	No se percibe una modificación susceptible de forma general	
	Presión atm	Se percibe un ligerísimo aumento ante el aumento de la presión.	
	Precipitación en mm	Se observa variabilidad, pero no un avance reseñable ante la precipitación.	
NO	*Vel. Viento (1m)*	Se percibe un clarísimo descenso ante el aumento de la velocidad. Remarcando el aumento de la contaminación ante situaciones sin viento.	
	Dir. Viento (1m)	No se percibe una gran variabilidad de forma general. <u>Pero en el caso de Manlleu se detecta una gran variabilidad según la dirección</u>. Hay un aumento en vientos de componente norte (de Barcelona a Manlleu).	(general)  (Manlleu) 
	Temperatura	En zonas de interior (Manlleu) se percibe un grandísimo descenso ante el aumento de temperatura (como hipótesis, menos necesidad de calefacción). A partir de una temperatura “mínima” (15º) se estabiliza el valor. En zonas de costa (Vilanova y Mataró) se percibe ligero descenso ante el aumento de temperatura.	(Manlleu)  (Mataró y Vilanova)

			
	Humedad	Se percibe un aumento ante el aumento de la humedad.	
	Presión atm	Se percibe un claro aumento ante el aumento de la presión	
	Precipitación en mm	Se observa variabilidad, ante lluvia ligera. Pero no un avance reseñable de la contaminación ante la precipitación.	
PM2.5	*Vel. Viento (1m)*	Se percibe un claro descenso ante el aumento de la velocidad. Remarcando el aumento de la contaminación ante situaciones sin viento y la estabilidad a partir de un calor de aumento. A partir de los 4m/s hay nulo descenso.	
	Dir. Viento (1m)	No se percibe una gran variabilidad de forma general. <u>Pero en el caso de Manlleu se detecta una gran variabilidad según la dirección</u>. Hay un aumento en vientos de componente norte-noreste(de Barcelona a Manlleu).	(general)  (Manlleu) 
	Temperatura	En zonas de interior	(Manlleu)

		(Manlleu) se percibe un grandísimo descenso ante el aumento de temperatura (como hipótesis, menos necesidad de calefacción). A partir de una temperatura “mínima” (15º) se estabiliza el valor. En zonas de costa (Vilanova) se percibe un descenso ante el aumento de temperatura.	 (Vilanova i la Geltrú) 
	Humedad	Se percibe un claro ascenso ante el aumento de la humedad.	
	Presión atm	Se observa un claro aumento ante el aumento de la presión.	
	Precipitación en mm	Se observa un claro descenso ante el aumento de la precipitación.	
PM10	*Vel. Viento (1m)*	Se percibe un claro descenso ante el aumento de la velocidad. Remarcando el aumento de la contaminación ante situaciones sin viento.	
	Dir. Viento (1m)	No se percibe una gran variabilidad de forma general. <u>Pero en el caso de Manlleu se detecta una variabilidad según la dirección</u>. Hay un aumento en vientos de componente norte-noreste (Barcelona-Manlleu).	(general)  (manlleu)

			
	Temperatura	Se percibe un grandísimo descenso ante el aumento de temperatura (como hipótesis, menos necesidad de calefacción) en zonas de interior (Manlleu) A partir de una temperatura “mínima” (15º) se produce un repunte de la contaminación. En zonas de costa se produce la evolución inversa, el aumento de temperatura precede al aumento de contaminación por PM10	(Manlleu)  (Mataró y Vilanova) 
	Humedad	No se perciben variaciones ante la variación de la humedad.	
	Presión atm	No se percibe una variación reseñable ante el aumento de la presión. Un ligero aumento.	
	Precipitación en mm	Se observa variabilidad, ante lluvia ligera. Pero no un avance reseñable de la contaminación ante la precipitación.	

10-Conclusiones

Como conclusión de este proyecto podemos constatar que es posible el realizar un lago de datos que procese los datos provenientes de sensores en un tiempo muy bajo para ofrecer resultados al usuario. Permitiéndonos cruzar los datos de polución y meteorología, para obtener nueva información a partir de ello.

El planteamiento de ingestar, almacenar, modificar, evaluar y posteriormente explotar los datos, provenientes de fuentes que aportan datos en batch es posible y realizable con las tecnologías actuales en el fascinante mundo del big data. Utilizando para ello todos los conocimientos obtenidos durante el estudio de este máster.

Entendiendo lo ofrecido como el desarrollo de una propuesta factible y funcional de entre las muchas posibles para la gestión de la información captada y entendiendo el trabajo con un lago de datos como algo evolutivo, que debe ir siendo refinado a medida que se va trabajando más con el dato. Para mejorar su velocidad, capacidad de almacenamiento y captación de más fuentes de datos para enriquecer más el dato en sí.

11-Evoluciones Futuras

Parafraseando otro ámbito: “las obras de arte nunca se acaban, solo se abandonan”, lo que nos lleva a reconocer que, por la limitación del tiempo disponible, no se ha podido llegar al nivel de perfeccionamiento que dejaría una talla perfecta en el pequeño diamante que ha resultado este TFM.

Con el espíritu que ha seguido esta obra, mencionaremos los puntos que se han reconocido de evolución, área por área, para que puedan ser retomados por cualquiera que quiera seguir los pasos y mejorar lo presente. No son necesarios para el correcto funcionamiento, pero toda mejora en lo presente redundará en la mejor eficiencia del sistema creado.

11.1-Entorno

Se ha de observar que todo el desarrollo se ha realizado en un ordenador personal que ha albergado todos los servidores a la vez, bajo Windows. Aun habiendo demostrado que es posible, se ha notado en algunos momentos la escasez de recursos.

- Realizar la implementación en diferentes máquinas, creando un sistema distribuido, para una ampliación de los recursos disponibles.
- Cambiar el sistema operativo donde se ha realizado la implementación de Windows a Linux, lo que daría muchos menos problemas en la implementación, siendo mucho más cercano al ser Linux el sistema operativo bajo el que corren los servidores implementados.

11.2-Docker

Dentro del área de Docker, siendo este el núcleo de la arquitectura en la que se ha desplegado la red de servidores las mejoras irían destinadas a mejorar la gestión de recursos.

- Eliminar servidores de la distribución que se han visto innecesarios para el desarrollo del proyecto: Zoekeeper,Kafka…

- Mejoras en la automatización de despliegue, generando un script para el autocopiado de la carpeta de configuración de Hadoop en el servidor de Nifi (esta se ha realizado de forma manual).

11.3-Nifi

Siendo el sistema de ingesta escogido y este habiendo cumplido todo lo que se esperaba de él toda mejora solo puede ir encaminada a la mejora en gestión, monitorización y seguridad.

- Implementación de un servidor Nifi Registry, para obtener un control de versiones del código y niveles de seguridad según usuarios.
- Despliegue de una versión de Nifi que obligue conexión por https para mejorar la seguridad.
- Implementación de sistema de comunicación con el usuario administrador en caso de fallo, para mejorar los tiempos de respuesta en caso de error.
- Implementación de sistema de monitorización preventivo, mediante código, ante presencia de datos inesperados.
- Cambio en el desarrollo para que todos los parámetros directamente se obtengan de un fichero hdfs, minimizando aún más la configuración, centrando toda en Hadoop.

11.4-Hadoop

Siendo el sistema donde albergamos toda la información, las mejoras deben ir encaminadas a mejorar los tiempos de respuesta y seguridad en el mismo.

- Aumento de número de datanodes para gestionar la replicación de ficheros, aumentando la seguridad.
- Aumento de recursos tanto para él/los datanodes, como para el namenode.
- Implementación de un Apache Ambari, para monitorizar el rendimiento del sistema Hadoop.

11.5-Hive

Siendo el sistema que hemos utilizado para dar forma a nuestros datos en el postprocesado, ha cumplido con los requisitos que buscábamos de velocidad, eficiencia y simplicidad. Aún así se podría dar una vuelta adicional.

- Implementación de postprocesado en Spark adicional, si queremos buscar algún dato o relación que pudiera ser excesivamente complejo de realizar en HIVE. Pudiendo ser también orquestado a través de Hue.
- Aumento del número de consultas realizadas y almacenadas, para poder abarcar nuevas necesidades que plantee el usuario.
- Depuración de las consultas, para buscar la máxima eficiencia de las mismas.
- Estudio para aumentar el número de campos ofertados en las consultas, para eliminar en lo posible la necesidad de creación de indicadores en la visualización y reducir al mínimo cálculo en la fase final. Por ejemplo, realizando la discretización del dato en HIVE.

- Establecimiento de particiones o buckets en las tablas para acelerar aún más su consulta. Dificulta la arquitectura y consultas. Pero devuelve el rédito en velocidad de acceso.
- Elaboración de la relación de datos para la cruzar predicciones meteorológicas. Se dispone del dato internamente,su carga está automatizada, pero se tendría que explotar.
- Argumentar un sistema para el reemplazo de los datos. En el caso de ejecución de las consultas de creación de los datos, los anteriores son machacados, con lo que se deberían ofrecer los datos "antiguos" hasta que se finalicen las consultas de creación de datos, imitando un sistema Lambda.

11.6-Power Bi

Como parte final de nuestro desarrollo, y siendo la parte que se presenta al usuario ha cumplido con las expectativas que se esperaba de la herramienta. Aún así es una parte, como todo, mejorable de cara a la experiencia del usuario.

- Aumento del número de diapositivas presentadas para dar un mayor espectro de análisis.
- Mejoras en la presentación de la información en los gráficos de geoposicionamiento, búsqueda de plugins más interesantes.
- Aporte de más información para el usuario, creación de más etiquetas, leyendas…

12-Dificultades halladas

No hay proyecto que salga en el primer intento perfecto. De la misma forma que hemos puesto de relevancia los resultados y conclusiones halladas también hay que dar importancia a todas las dificultades que se han sucedido en el desarrollo y como se han resuelto para llegar a la solución final.

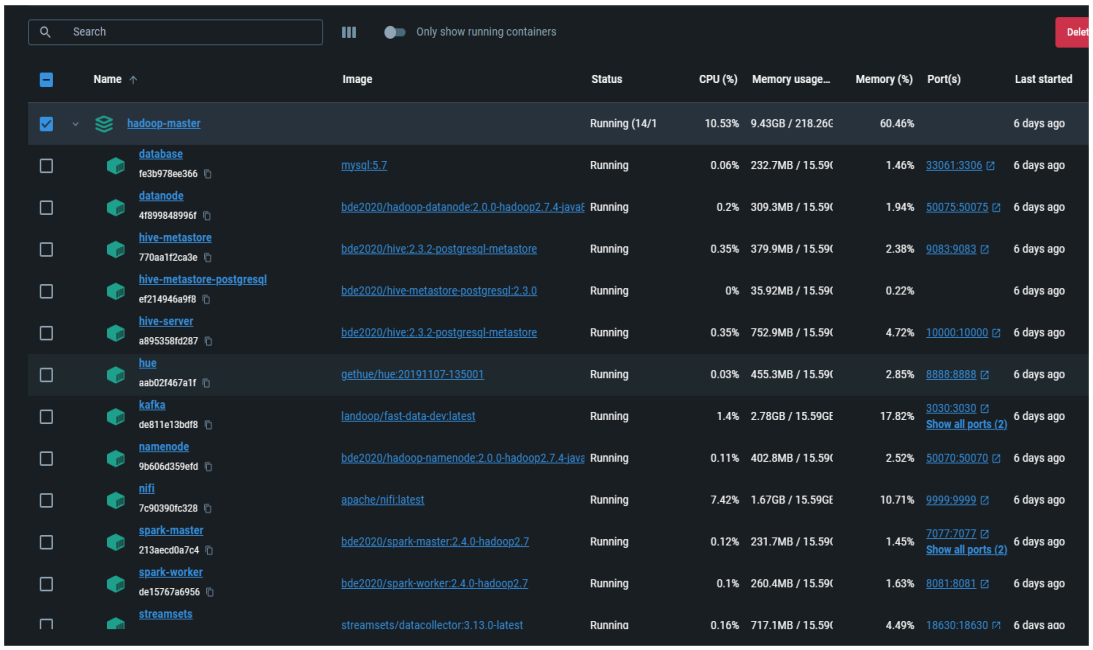
Es importante mencionarlas, de cara a evitar que los mismos problemas sean reproducidos. En caso de ocurrir los mismos o similares, se tenga una guía de cómo obrar ante ellos.

Siendo una parte, la resolución de estos problemas, que ha consumido mucho tiempo en el desarrollo es necesario el darles cabida en esta memoria. Estructurando por área donde han ocurrido estos problemas.

12.1-Herramienta Docker

En la configuración de la herramienta Docker es donde se ha registrado la mayor parte de los problemas del proyecto. En la configuración de la red de máquinas que hospedan las aplicaciones que necesitamos.
Estas dificultades son críticas, ya que sin solucionarlas no podría desarrollarse propiamente y detallamos las que hemos tenido, pudiendo haberle dado solución a todas ellas, afortunadamente.
Principalmente cada uno de estos fallos ha necesitado, como mínimo, alteraciones en el fichero Docker-compose.yml (la plantilla de despliegue de los servidores. Lo que requiere utilizar el comando *Docker-compose down* y *Docker-compose up*, renovando todo el despliegue.

- No visualización de las maquinas entre si.
    - Inicialmente se intentó realizar despliegues por separado de los servidores necesarios. Siendo puristas e intentando desplegar las versiones más actualizadas. Topamos con el problema de no visibilización entre los diferentes servidores. A la hora de la comunicación esto no era posible.
    - Los servidores internamente no saben que hay un servidor en una red externa, están encapsulados.
    - Esto no hace imposible la comunicación ya que podríamos hacer que un servidor, para comunicarse, buscase externamente y recuperase la información que necesite. Pero esto nos impone una penalización en eficiencia y en el tiempo que nos consumirá para toda la configuración adicional y pruebas (del que no disponemos por las limitaciones del proyecto)
    - Por ello hemos decidido crear una distribución dentro de Docker. Una red de servidores. Mirando la página oficial buscamos en la página oficial diferentes opciones para ver la que nos conviene más. (que contenga nifi, hadoop, hue, hive…)
      De las opciones disponibles probamos:
        - big-data-europe (#12.1)
        - marcel-jan config (#12.2)
        - julio-lopez config(#12.3)
    - Optamos por la opción del especialista julio-lopez y procedimos a adaptarla a nuestras necesidades, modificando los ficheros de configuración y aplicando la distribución. Dándonos el resultado deseado:



(48.Red de servidores utilizados en nuestro proyecto en el entorno docker)

- No tener persistencia de la información (desaparece toda tras reinicio de la herramienta)
    - Tras cada reinicio del servidor Docker (planeado o no). Los datos desaparecen, no siendo persistentes.
      Esto ocurre porque Docker, internamente no está pensado para la persistencia

interna. Para que los datos persistan estos tienen que estar almacenados externamente al sistema Docker, en nuestro sistema de archivos. Para ello debemos utilizar el sistema de Docker para “bindear” carpetas, creando volúmenes accesibles tanto desde dentro del sistema Docker, estando mapeadas, como externamente.

- Lo más recomendable es realizarlo en el despliegue, modificando el fichero Docker-compose.yml indicando el volumen a “bindear” en la sección “volumes” del servidor. Indico en las capturas los volúmenes creados:

```
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop2.7.4-java8
    container_name: namenode
    volumes:
      - E:\HadoopData\Hdfs\NameNode:/hadoop/dfs/name
```

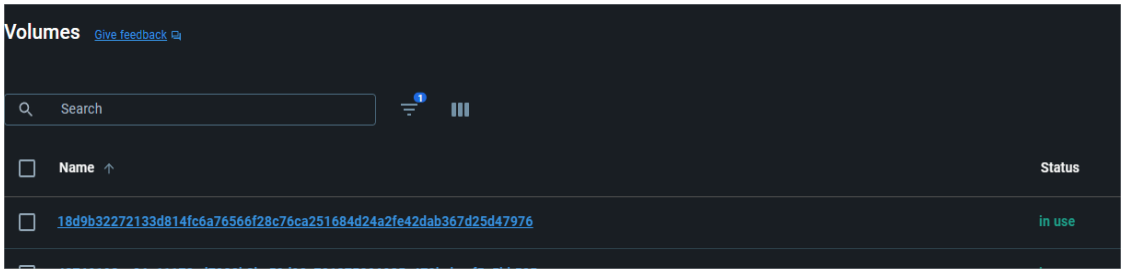
```
  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop2.7.4-java8
    container_name: datanode
    volumes:
      - E:\HadoopData\Hdfs\DataNode:/hadoop/dfs/data
      - ./bank:/bank
    env_file:
      - ./hadoop-hive.env
```

```
  hue:
    image: gethue/hue:20191107-135001
    hostname: hue
    container_name: hue
    dns: 8.8.8.8
    ports:
      - "8888:8888"
    volumes:
      - ./hue-overrides.ini:/usr/share/hue/desktop/conf/hue.ini
```

```
  database:
    image: mysql:5.7
    container_name: database
    ports:
      - "33061:3306"
    command: --init-file /data/application/init.sql
    volumes:
      - E:\HadoopData\MySql\data:/var/lib/mysql
```

(49.Capturas de los diferentes volúmenes configurados en el fichero docker-compose.yml)

Puede configurarse dentro del sistema Docker.



(50.Ejemplo de volumen creado internamente en docker).

Pero es más cómodo, una vez se tiene claro, realizarlo externamente en el fichero *Docker-compose.yml* Facilitando despliegues posteriores.

Es necesario tras cada modificación en el fichero Docker-compose ejecutar los comandos *docker-compose down* y *docker-compose-down* (producirá un redespliegue de los servidores.

- Los servidores de hadoop no permiten escritura
    - En la escritura en el sistema hadoop no permite escribir tras recibir peticiones externas (por ejemplo escrituras desde NIFI).

      Las causas pueden ser múltiples, en nuestro caso no estaba bien configurada las carpetas destino donde se escribirían los datos (es decir, no había espacio para ello).
    - Creando un volumen de escritura externo para el datanode y namenode se subsana el problema (mencionado en el punto anterior).
- El servidor hadoop no puede escribir datos por no poder crear las réplicas suficientes.
    - Al intentar escribir en los servidores hadoop, esto no es posible y registra el fallo en el log:

      org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicy: Failed to place enough replicas, still in need of 3 to reach 3 (unavailableStorages=[], storagePolicy=BlockStoragePolicy{ALL_SSD:12, storageTypes=[SSD], creationFallbacks=[DISK], replicationFallbacks=[DISK]}, newBlock=true) For more information, please enable DEBUG log level on org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicy and org.apache.hadoop.net.NetworkTopology

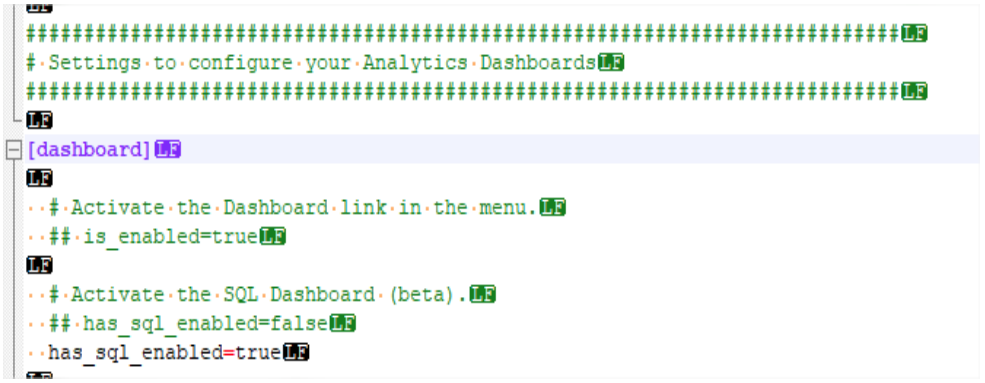
      (51.Ejemplo de fallo registrado en el log)
    - Hemos usado la referencia en (#12.4) para la resolución de la incidencia. Siendo un punto de configuración adicional dentro del servidor
- El servidor Hue no se activa correctamente
    - Al levantarse el servidor se puede recibir un fallo indicando que faltan las tablas de configuración.

      Esto es crítico, ya que Hue no podrá arrancar sin ellas.
    - Se soluciona migrando las tablas de usuario al servidor mysql. Se utiliza el comando */usr/share/hue/build/env/bin/hue migrate* en el servidor mysql. Tras un reinicio ya funcionará correctamente
- El servidor hue no visualiza el sistema hadoop.
    - Los parámetros de configuración del servidor hue no son correctos.

      Se debe modificar el fichero hue.ini con los parámetros necesarios.
        - Incorporando la ruta correcta.

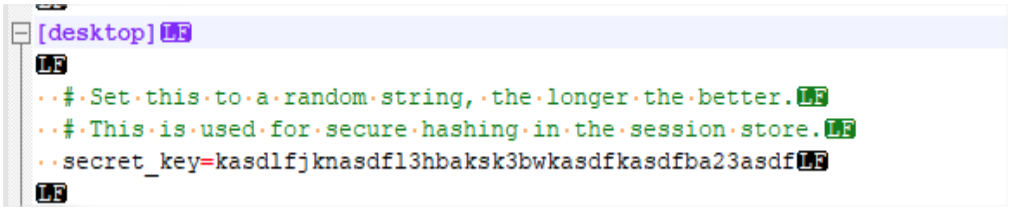
(52.Configuración en hue.ini para conexión con hdfs)

- Indicando que el sql está habilitado



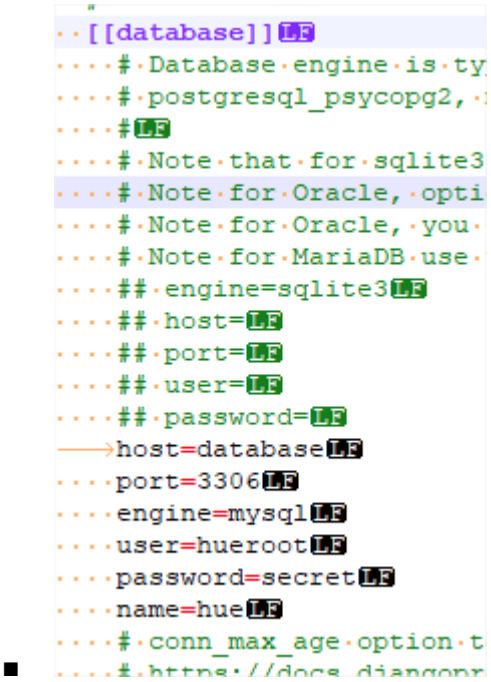
(53.Configuración en hue.ini para conexión sql)

- Indicando la secret-key



(54.Configuración en hue.ini para tener la obligatoria secret-key)

- Indicando los datos correctos del servidor mysql utilizado



(55.Configuración en hue.ini para conexión con servidor mysql)

- El problema radica en que cada modificación en el fichero hue.ini se perderá tras cada reinicio. No hay persistencia.

  Para ello creamos un fichero de modificaciones hue-overrides.ini, lo almacenaremos en una carpeta que "bindearemos" en el despliegue de servidores e indicaremos que este fichero machacará el hue.ini en el servidor. Haciendo efectivos los cambios indicados.

- Modificación en el fichero Docker-compose.yml:

(56.Configuración en docker-compose.yml para machacar el fichero hue.ini con hue-overrides.ini)

- Como soporte se ha usado la información hallada en (#12.5) y (#12.6)

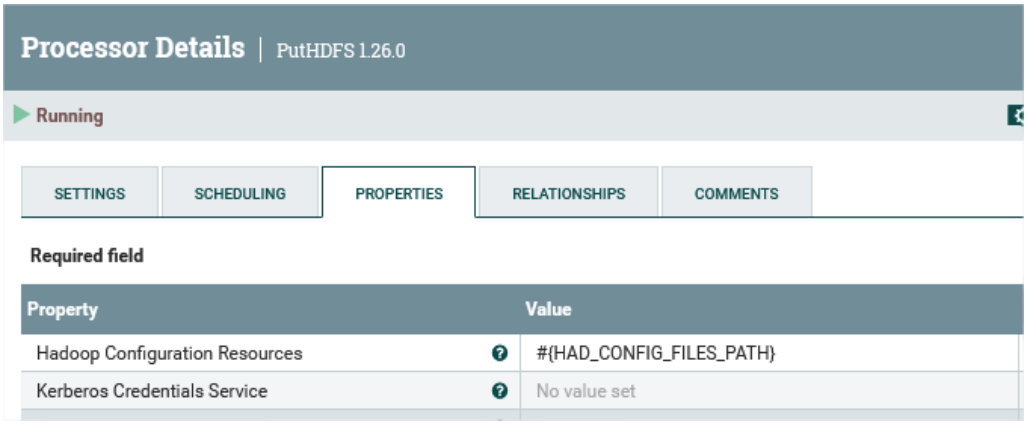
- Problemas al levantar un servidor (el que sea), por no estar el puerto disponible.
  - Este es un error muy molesto, ya que inicialmente la máquina en cuestión no se levantará por sí sola. Se tiene que revisar dentro de los logs de la misma el fallo.
  - Una vez detectado hay que verificar que ninguna máquina creada mediante docker comparte puertos (algo que nos ocurrió entre la maquina de nifi y la de hue) y modificarlos. Es importante controlar que no sea una aplicación externa la que ya

tenga ocupado el puerto.

12.2-Entorno Nifi

Siendo Nifi una herramienta muy trillada y con la que se ha trabajado anteriormente se esperaban pocos problemas:

- Visualización de entorno y recuperación de datos HDFS.
    - Se hace necesario copiar ficheros de configuración presentes en el sistema HDFS en el servidor, sino la comunicación entre los procesadores HDFS de Nifi y los servidores no habrá comunicación.
    - Los ficheros de configuración son *hdfs-site.xml* y *core-site.xml* (en la ruta /etc/hadoop/ dentro del servidor namenode).
      Deben ser copiados en una ruta dentro del servidor NIFI (en nuestro caso hemos escogido */etc/hadoop/*).
      Posteriormente será referenciada dentro de los procesadores nifi (usamos parámetro para ello).



(57.Invocación de parámetros contextuales para la conexión con un servidor hdfs para hacer operaciones PUT)

- El formato de los datos recuperados no es uniforme. Recibiendo respuestas en .json, pero con formatos muy diferentes.
    - Unas consultas escalan en número de columnas recibidas en el mismo registro (consultas de contaminación), otras devuelven valores como array (predicciones), otras escalan como número de registros (climatología).
    - Pudiendo gestionar este problema en el entorno Hive o Nifi, se ha optado preprocesar en el entorno Nifi para uniformizar el dato y simplificar el uso de HIVE.
- Las fuentes de datos imponen límites en los registros recuperados, de tamaño.
    - Se opta por un sistema de recuperación de datos en ventanas temporales pequeñas para sortear el problema.

12.3-Entorno Hadoop

En el repositorio de ficheros Hadoop no se han detectado problemas per se. Siendo todos heredados por la herramienta Docker. Una vez resueltos los problemas de configuración en Docker, este ha funcionado con normalidad. Siendo un problema de conectividad entre entornos.

12.4-Entorno Hive

El entorno HIVE ha reportado pocos problemas. Siendo estos:

- Desaparición de las consultas almacenadas.
    - Solucionado al establecer un repositorio permanente externo a Docker (se reseteaba tras cada desconexión) para que la base de datos mysql que las alberga no fuera eliminada volviéndose persistente.
        - Tratado en el punto sobre las dificultades de Docker.
- Problemas con los formatos de datos.
    - No existe un tipo de dato en HIVE date/time. Existe date y timestamp.Para solucionarlo se ha tenido que refinar más las consultas utilizadas y tenerlo en cuenta de cara a la utilización en Power Bi.
    - Como guía tenemos la documentación oficial (#12.7)
- Problemas con manejo de tablas externas.
    - Se debe ser muy escrupuloso y preciso con las rutas utilizadas. En caso de haber un fallo, los datos se moverán a direcciones que no tenemos controladas y creeremos que simplemente han desaparecido.
- Problema con el volumen de las consultas.
    - Sobre el papel HIVE puede consultar cantidades ingentes de datos. manejando Terabytes con soltura. Pero hemos constatado que hay peros.
      Si los datos están en aislados en múltiples ficheros (un registro por fichero .json o.csv por ejemplo) al tener que recorrerlos en el sistema HDFS esto implica recuperar fichero a fichero, lo que nos produce que al realizar consultas con JOINS estas consuman mucho tiempo (constatamos con horror tiempos superiores a una hora en consultas con 1M de registros con múltiples INNER JOIN). Llegando en un caso a dar problemas de memoria.
    - Después de repasar el sistema de ficheros y constatar que no era problema de recursos se llegó a una solución realizando una compactación de los registros. Mediante postprocesado se insertan todos los registros nuevamente en una tabla destino. Lo que produce que se compactará de n ficheros a un número mucho menor y manejable, pero de mayor tamaño.
        - Esto produjo que de tiempos superiores a 1 hora por consulta se bajó a menos de 1 segundo.
    - Se consultó la referencia (#12.8) para mejorar más el entorno HIVE.

12.5-Entorno Power Bi

En la herramienta Power Bi podemos admitir que se han sucedido pocas incidencias.

- No conseguir inicialmente una conexión estable contra el entorno HDFS.
    - Solucionando mediante el establecimiento de una conexión ODBC para la comunicación de datos. Como guía se ha utilizado (#12.9)
- Dificultad en mantener ordenadas las jerarquías de cara a la visualización.

    - Solucionado tras consultar, realizando ordenaciones adicionales de las columnas. (#12.10)
- Necesidad de crear una tabla date/time con aumentos programados (en este caso de 1 hora).
    - Para ello se ha consultado en comunidades de expertos que han guiado en la creación de la fórmula DAX que se adapta mejor a nuestros requerimientos. (#12.11)

13-Agradecimientos

A mi tutor Juan Carlos Castro Robles, por la paciencia demostrada, la gran disponibilidad para las múltiples reuniones que hemos tenido, saber guiarme durante la elaboración del proyecto y el apoyo mostrado.

A Jaime Pallarés Bel, por el apoyo mostrado y la inspiración inicial aportada.

A toda la comunidad de Stack Overflow, por las múltiples dudas solucionadas en el desarrollo del proyecto.

Al canal de youtube agus100cia (#13.1) sobre Big Data por ayudarme a aumentar mis conocimientos en HIVE.

A la página de transparencia de la Generalitat y la sección de open data de meteocat por facilitar y aportar toda una plataforma de open data de la que hemos nutrido el proyecto. Sin la cual no habría sido posible.

Para finalizar y muy especialmente a Julio Lopez Nuñez (#13.2), autor de la distribución inicial de Docker realizada a partir de la cual realizamos la configuración.

14-Glosario

- ZBE: Zona de bajas emisiones.
- BOE: Boletín oficial del estado.
- OMS: Organización mundial de la salud.
- WHO: World health association - Organización mundial de la salud.
- Meteocat: Servicio Meteorológico de Catalunya
- Xema: Red de estaciones meteorológicas automáticas (xarxa d'estacions meteorologiques automatiques).
- HDFS: Hadoop file system. Un sistema de ficheros distribuido utilizado en este proyecto.

15-Bibliografía

Indicamos las referencias bibliográficas indicando el lugar de su aparición siguiendo el formato (#[núm. apartado].[núm. de aparición en el apartado]) para facilitar su búsqueda y relación.

(#1.1) - BOE Real decreto 01/2023 sobre contaminantes - (consultado en 04/2024)
https://www.boe.es/buscar/doc.php?id=BOE-A-2023-2026
(#1.2) - Referencia a la contaminación atmosférica de la OMS - (consultado en 04/2024)
https://www.who.int/es/health-topics/air-pollution#tab=tab_1
(#1.3) - Predicting air quality (and pollution) with Grafana - (consultado en 04/2024)
https://grafana.com/events/grafanacon/2021/air-quality-grafana/
(#1.4) - Evolution of Gaseous and Particulate Pollutants in the Air: What Changed after Five Lockdown Weeks at a Southwest Atlantic European Region (Northwest of Spain) Due to the SARS-CoV-2 Pandemic? -
Autores:Jorge Moreda-Piñeiro, María Fernández-Amado, Paula Costa-Tomé, Nuria Gallego-Fernández, María Piñeiro-Iglesias, Purificación López-Mahía, Soledad Muniategui-Lorenzo
Medio: Atmosphere (publicado 2021). Texto completo.
https://www.mdpi.com/2073-4433/12/5/562
(#1.5) - Estudio de la evolución de contaminantes atmosféricos basado en variables meteorológicas
Autor: Alejandro López Gómez
Medio: Universidad de Huelva y Universidad Internacional de Andalucía (2023). Texto completo
http://www.uhu.es/mecofin/documents/docencia/tfm/20222023/MECOFIN_20222023_tfm-LGA_EDLE.pdf
(#1.7) - An integrated analysis of air pollution and meteorological conditions in Jakarta
Autor: Teny Handhayani
Medio: Scientific reports (2023) Texto completo
https://www.nature.com/articles/s41598-023-32817-9
(#1.8) - Predicting Air Quality from Measured and Forecast Meteorological Data: A Case Study in Southern Italy
Autores: Andrea Tateo, Vincenzo Campanaro, Nicola Amoroso, Loredana Bellantuono, Alfonso Monaco, Ester Pantaleo, Rosaria Rinaldi, Tommaso Maggipinto
Medio: Atmosphere (publicado 2023). Texto completo
https://www.mdpi.com/2073-4433/14/3/475
(#1.9) - Data analysis and mining of the correlations between meteorological conditions and air quality: A case study in Beijing
Autores: Xiaoyu Qi, Gang Mei, Salvatore Cuomo, Chun Liu, Nengxiong Xu
Medio: Science Direct (publicado en 2021). Texto completo
https://www.sciencedirect.com/science/article/abs/pii/S2542660519301969
(#1.10) - Plataforma miteco - (consultado en 04/2024)
https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/evaluacion-datos/eval.html
(#1.11) - Sistema Caliope - (consultado en 05/2024)
http://www.bsc.es/caliope/es
(#1.12) - Metodología Lean - (consultado en 06/2024)
https://www.atlassian.com/es/agile/project-management/lean-methodology
(#2.1) - Detalle sobre tipos de contaminantes - (consultado en 06/2024)
https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/health-impacts/types-of-pollutants
(#2.2) - BOE Real decreto 01/2023 sobre contaminantes - (consultado en 04/2024)
https://www.boe.es/buscar/doc.php?id=BOE-A-2023-2026
(#2.3) - Relación del metadato meteorológico a analizar en la red XEMA - (consultado en 04/2024)
https://analisi.transparenciacatalunya.cat/Medi-Ambient/Metadades-variables-meteorol-giques/4fb2-n3yi/data_preview
(#3.1) - Sistema Caliope - (consultado en 05/2024)
http://www.bsc.es/caliope/es
(#3.2) - Estudios de herramientas ETL - (consultado en 04/2024)
https://opensistemas.com/herramientas-etl-mas-usadas/ https://www.modus.es/herramientas-etl-mas-usadas/
(#3.3) - Estudios sobre el almacenamiento - (consultado en 04/2024)
https://www.modus.es/herramientas-etl-mas-usadas/
(#4.1) - Página de descarga de Docker Desktop - (consultado en 04/2024)
https://www.docker.com/products/docker-desktop/
(#4.2) - Página de Git de descarga de distribución de Docker de Julio Lopez

https://github.com/juliopez/Hadoop - (consultado en 04/2024)
(#5.1) - Detalle sobre contaminantes recuperados de la plataforma OpenData - (consultado en 04/2024)
https://analisi.transparenciacatalunya.cat/Medi-Ambient/Qualitat-de-l-aire-als-punts-de-mesurament-autom-t/tasf-thgu/about_data
(#5.2) - Detalle sobre meteorología recuperada de la plataforma OpenData - (consultado en 04/2024)
https://analisi.transparenciacatalunya.cat/Medi-Ambient/Dades-meteorol-giques-de-la-XEMA/nzvn-apee/about_data
(#6.1) - Página de inicio de Apache Hadoop - (consultado en 04/2024)
https://hadoop.apache.org/
(#7.1) - Página principal de Apache HIVE - (consultado en 05/2024)
https://hive.apache.org/
(#7.2) - Página principal de Apache Hue - (consultado en 05/2024)
https://gethue.com/
(#12.1) - Big Data Europe - (consultado en 04/2024)
https://github.com/big-data-europe/Docker-hadoop
(#12.2) - Docker multi-container environment with Hadoop, Spark and Hive - (consultado en 04/2024)
https://github.com/Marcel-Jan/Docker-hadoop-spark
(#12.3) - Hadoop / Docker-Compose by @Juliopez - (consultado en 04/2024)
https://github.com/juliopez/Hadoop
(#12.4) - Failed to place enough replicas, still in need of 3 - (consultado en 04/2024)
https://community.cloudera.com/t5/Support-Questions/URGENT-case-Failed-to-place-enough-replicas-still-in-need-of/m-p/339765
(#12.5) - Hue access to HDFS: bypass default hue.ini? - (consultado en 04/2024)
https://stackoverflow.com/questions/57116402/hue-access-to-hdfs-bypass-default-hue-ini
(#12.6) - Levantando HUE en AWS (Resolviendo problema) + Docker - (consultado en 05/2024)
https://www.youtube.com/watch?v=Ck4sRPa0o24
(#12.7) - Troubleshooting Errors and Exceptions in Hive Jobs — Qubole Data Service documentation - (consultado en 04/2024)
https://docs.qubole.com/en/latest/troubleshooting-guide/hive-ts/troubleshoot-hive.html
(#12.8) - LanguageManual UDF - Apache Hive - Apache Software Foundation - (consultado en 04/2024)
https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-StringOperators
(#12.9) - Visualización de big data en tiempo real| ¿Cómo conectar Hive y Power Bi? - Video - (consultado en 05/2024)
https://www.youtube.com/watch?v=tOcfI7bGEdI
(#12.10) - Solved: Date Hierarchy is not in order - Microsoft Fabric Community - (consultado en 05/2024)
https://community.fabric.microsoft.com/t5/Desktop/Date-Hierarchy-is-not-in-order/td-p/2700911
(#12.11) - How to create a Date/Time Table with 1-hour increments on each row using DAX - (consultado en 05/2024)
https://www.vahiddm.com/post/creating-a-date-time-table-using-dax
(#13.1) - Canal de youtube sobre Big Data agus100cia - (consultado en 04/2024)
https://www.youtube.com/@agus100cia
(#13.2) - Blog de Julio Lopez Nuñez - (consultado en 04/2024)
https://juliopezblog.wordpress.com/

16-Anexos

Sección docker

Contenido fichero docker-compose.yml
version: '3' services: namenode:

```
  image: bde2020/hadoop-namenode:2.0.0-hadoop2.7.4-java8
  container_name: namenode
  volumes:
    - E:\HadoopData\Hdfs\NameNode:/hadoop/dfs/name
  environment:
    - CLUSTER_NAME=test
  env_file:
    - ./hadoop-hive.env
  ports:
    - "50070:50070"
  networks:
    net_pet:
      ipv4_address: 172.27.1.5

datanode:
  image: bde2020/hadoop-datanode:2.0.0-hadoop2.7.4-java8
  container_name: datanode
  volumes:
    - E:\HadoopData\Hdfs\DataNode:/hadoop/dfs/data
    - ./bank:/bank
  env_file:
    - ./hadoop-hive.env
  environment:
    SERVICE_PRECONDITION: "namenode:50070"
  depends_on:
    - namenode
  ports:
    - "50075:50075"
  networks:
    net_pet:
      ipv4_address: 172.27.1.6

hive-server:
  image: bde2020/hive:2.3.2-postgresql-metastore
  container_name: hive-server
  env_file:
    - ./hadoop-hive.env
  environment:
    HIVE_CORE_CONF_javax_jdo_option_ConnectionURL: "jdbc:postgresql://hive-metastore/metastore"
    SERVICE_PRECONDITION: "hive-metastore:9083"
  ports:
    - "10000:10000"
  depends_on:
    - hive-metastore
  networks:
    net_pet:
      ipv4_address: 172.27.1.7

hive-metastore:
  image: bde2020/hive:2.3.2-postgresql-metastore
  container_name: hive-metastore
  env_file:
    - ./hadoop-hive.env
  command: /opt/hive/bin/hive --service metastore
  environment:
    SERVICE_PRECONDITION: "namenode:50070 datanode:50075 hive-metastore-postgresql:5432"
  ports:
    - "9083:9083"
  depends_on:
    - hive-metastore-postgresql
  networks:
    net_pet:
```

```
      ipv4_address: 172.27.1.8

hive-metastore-postgresql:
  image: bde2020/hive-metastore-postgresql:2.3.0
  container_name: hive-metastore-postgresql
  depends_on:
    - datanode
  networks:
    net_pet:
      ipv4_address: 172.27.1.9

spark-master:
  image: bde2020/spark-master:2.4.0-hadoop2.7
  container_name: spark-master
  ports:
    - 8080:8080
    - 7077:7077
  environment:
    - CORE_CONF_fs_defaultFS=hdfs://namenode:8020
  env_file:
    - ./hadoop-hive.env
  networks:
    net_pet:
      ipv4_address: 172.27.1.10

spark-worker:
  image: bde2020/spark-worker:2.4.0-hadoop2.7
  container_name: spark-worker
  depends_on:
    - spark-master
  environment:
    - SPARK_MASTER=spark://spark-master:7077
    - CORE_CONF_fs_defaultFS=hdfs://namenode:8020
    - HIVE_CORE_CONF_javax_jdo_option_ConnectionURL=jdbc:postgresql://hive-metastore/metastore
  ports:
    - 8081:8081
  env_file:
    - ./hadoop-hive.env
  networks:
    net_pet:
      ipv4_address: 172.27.1.11

zeppelin:
  image: apache/zeppelin:0.8.0
  container_name: zeppelin
  ports:
    - 19090:8080
  networks:
    net_pet:
      ipv4_address: 172.27.1.12

hue:
    image: gethue/hue:20191107-135001
    hostname: hue
    container_name: hue
    dns: 8.8.8.8
    ports:
    - "8888:8888"
    volumes:
      - ./hue-overrides.ini:/usr/share/hue/desktop/conf/hue.ini
    depends_on:
    - "database"
```

```
    networks:
      net_pet:
        ipv4_address: 172.27.1.13

database:
    image: mysql:5.7
    container_name: database
    ports:
        - "33061:3306"
    command: --init-file /data/application/init.sql
    volumes:
        - E:\HadoopData\MySql\data:/var/lib/mysql
        - ./init.sql:/data/application/init.sql
    environment:
        MYSQL_ROOT_USER: hueroot
        MYSQL_ROOT_PASSWORD: secret
        MYSQL_DATABASE: hue
        MYSQL_PASSWORD: secret
        MYSQL_USER: hueroot
    networks:
      net_pet:
        ipv4_address: 172.27.1.14

zookeeper:
  hostname: zookeeper
  container_name: zookeeper
  image: bitnami/zookeeper:latest
  environment:
    - ALLOW_ANONYMOUS_LOGIN=yes
  ports:
    - "2181:2181"
  networks:
    net_pet:
      ipv4_address: 172.27.1.15

kafka:
  image: landoop/fast-data-dev:latest
  container_name: kafka
  ports:
    - "9092:9092"
    - "3030:3030"
  environment:
    KAFKA_ADVERTISED_HOST_NAME: 172.27.1.16
    KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
  networks:
    net_pet:
      ipv4_address: 172.27.1.16

streamsets:
  image: streamsets/datacollector:3.13.0-latest
  container_name: streamsets
  ports:
    - "18630:18630"
  networks:
    net_pet:
      ipv4_address: 172.27.1.17

nifi:
  image: apache/nifi:latest
  container_name: nifi
  ports:
    - 9999:9999
```

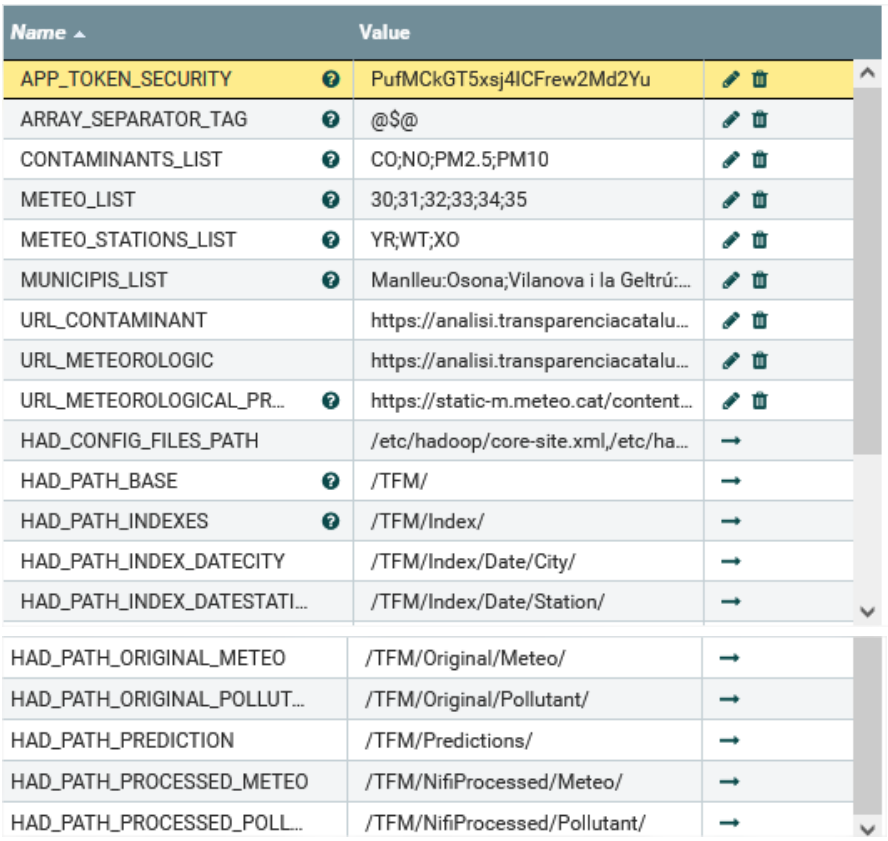
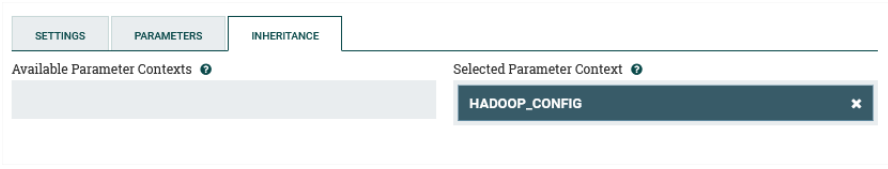
```
    environment:
      - NIFI_WEB_HTTP_PORT=9999
      - NIFI_CLUSTER_IS_NODE=true
      - NIFI_CLUSTER_NODE_PROTOCOL_PORT=8082
      - NIFI_ZK_CONNECT_STRING=zookeeper:2181
      - NIFI_ELECTION_MAX_WAIT=1 min
      - NIFI_SENSITIVE_PROPS_KEY=muIzsOYZhDI4
    networks:
      net_pet:
        ipv4_address: 172.27.1.18

networks:
  net_pet:
    ipam:
      driver: default
      config:
        - subnet: 172.27.0.0/16
```

Sección NIFI

(mostraremos los parámetros del entorno y detallaremos todos los procesadores del proceso principal)

Parámetros de contexto	
Hadoop config	SETTINGS / PARAMETERS / INHERITANCE

Name	Value
HAD_CONFIG_FILES_PATH	/etc/hadoop/core-site.xml,/etc/had...
HAD_PATH_BASE	/TFM/
HAD_PATH_INDEXES	/TFM/Index/
HAD_PATH_INDEX_DATECITY	/TFM/Index/Date/City/
HAD_PATH_INDEX_DATESTATION	/TFM/Index/Date/Station/
HAD_PATH_ORIGINAL_METEO	/TFM/Original/Meteo/
HAD_PATH_ORIGINAL_POLLUTA...	/TFM/Original/Pollutant/
HAD_PATH_PREDICTION	/TFM/Predictions/
HAD_PATH_PROCESSED_METEO	/TFM/NifiProcessed/Meteo/
HAD_PATH_PROCESSED_POLLUT...	/TFM/NifiProcessed/Pollutant/

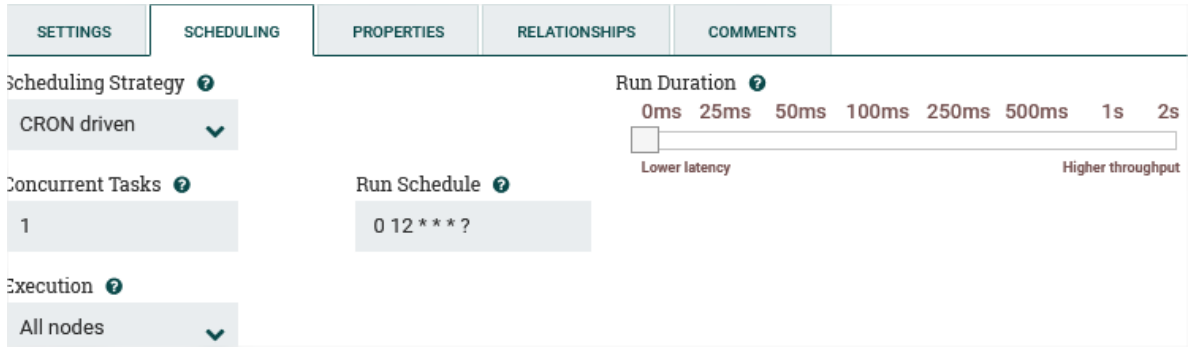
Basic import config	
Relación de herencia de basic import config a hadoop config	

Disparador del proceso

Property	Value
Data Format	**Text**
Unique FlowFiles	**false**
Custom Text	#{MUNICIPIS_LIST}
Character Set	**UTF-8**
Mime Type	No value set
arrSeparator	#{ARRAY_SEPARATOR_TAG}
dateEnd	${now():toNumber():minus(86400000):toDate():format("...
meteoConditions	#{METEO_LIST}
meteoStations	#{METEO_STATIONS_LIST}
nowFmt	${now():format("dd/MM/yyyy HH:mm:00")}
pollutants	#{CONTAMINANTS_LIST}
timeStampSecond	${now():format("dd/MM/yyyy HH:mm:00.000'Z'"):toDate(...

dateEnd:
${now():toNumber():minus(86400000):toDate():format("yyyy-MM-dd")}

timeStampSecond:
${now():format("dd/MM/yyyy HH:mm:00.000'Z'"):toDate("dd/MM/yyyy HH:mm:00.000'Z'"):toNumber()}



Grupo de proceso importPollutionData *(incluimos solo las secciones configuradas)*	
SplitContent	**Property** / **Value** **Byte Sequence Format**: **Text** **Byte Sequence**: ; **Keep Byte Sequence**: **false** **Byte Sequence Location**: **Trailing**
get City Region FromContext	**Enable Unix Lines Mode**: **false** **Include Capture Group 0**: **false** **Enable repeating capture group**: **false** Enable named group support: false municipiRegion: (?s)(.*)
getCity	**Property** / **Value** Delete Attributes Expression: No value set **Store State**: **Do not store state** Stateful Variables Initial Value: No value set **Cache Value Lookup Cache Size**: **100** municipi: ${municipiRegion:substringBefore(":")}
putPollutantsContext	**Property** / **Value** **Replacement Strategy**: **Regex Replace** **Search Value**: **(?s)(^.*$)** **Replacement Value**: **${pollutants}** **Character Set**: **UTF-8** **Maximum Buffer Size**: **1 MB** **Evaluation Mode**: **Line-by-Line** Line-by-Line Evaluation Mode: All

SplitPollutants

Property	Value
Byte Sequence Format	Text
Byte Sequence	;
Keep Byte Sequence	false
Byte Sequence Location	Trailing

etPollutantFromContext

Enable Unix Lines Mode	false
Include Capture Group 0	false
Enable repeating capture group	false
Enable named group support	false
pollutant	(?s)(.*)

FetchStartDateHDFS

Property	Value
Hadoop Configuration Resources	#{HAD_CONFIG_FILES_PATH}
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
HDFS Filename	#{HAD_PATH_INDEX_DATECITY}${municipi}_${pollutant}
Compression codec	NONE

setDateTwoDaysAgo

Property	Value
Replacement Strategy	Regex Replace
Search Value	(?s)(^.*$)
Replacement Value	${now():toNumber():minus(172800000):toDate():format(...
Character Set	UTF-8
Maximum Buffer Size	1 MB
Evaluation Mode	Line-by-Line
Line-by-Line Evaluation Mode	All

ReplacementValue:
${now():toNumber():minus(172800000):toDate():format("yyyy-MM-dd")}

getStartDateFromContext

Enable Unix Lines Mode	false
Include Capture Group 0	false
Enable repeating capture group	false
Enable named group support	false
dateStart	(?s)(.*)

prepareRequestUrl

Property	Value
Delete Attributes Expression	No value set
Store State	Do not store state
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	100
urlParam	?$where=...

urlParam:
?$where=
${pollutant:isEmpty():ifElse("",${pollutant:replace(";","' OR contaminant='"):append("')"):prepend("(contaminant='")})}
${municipi:isEmpty():ifElse("",${municipi:prepend(" AND municipi='"):append("' ")})}
${dateStart:isEmpty():ifElse("",${dateStart:prepend(" AND data BETWEEN'"):append("' AND '"):append(${dateEnd}):append("' ")})}

getDataHttp

Property	Value
HTTP Method	**GET**
HTTP URL	**#{URL_CONTAMINANT}${urlParam}**
HTTP/2 Disabled	**False**
SSL Context Service	No value set
Response FlowFile Naming Strategy	**RANDOM**
Response Header Request Attributes Enabled	false
Response Redirects Enabled	**True**
X-App-Token	#{APP_TOKEN_SECURITY}

SplitRegisters By day

Property	Value
JsonPath Expression	**$.***
Null Value Representation	**empty string**
Max String Length	**20 MB**

Proseguimos con la línea secundaria de datos no procesados y anotación de índice

setFileName

Property	Value
Delete Attributes Expression	No value set
Store State	**Do not store state**
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	**100**
filename	${municipi}_dateStart${dateStart}_dateEnd${dateEnd}

insertOriginalPollutantHDFS

Property	Value
Hadoop Configuration Resources	#{HAD_CONFIG_FILES_PATH}
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
Directory	**#{HAD_PATH_ORIGINAL_POLLUTANT}**
Conflict Resolution Strategy	**replace**
Writing Strategy	**Simple write**
Block Size	No value set

Script getDate

Property	Value
Script Engine	**Groovy**
Script File	No value set
Script Body	import org.apache.commons.io.IOUtils;...
Module Directory	No value set

Código Groovy:

```
import org.apache.commons.io.IOUtils;
import java.nio.charset.StandardCharsets;
//get flowfile, if not, abort
flowFile = session.get();
if(!flowFile)return;
// Cast a closure with an inputStream parameter to InputStreamCallback
session.read(flowFile, {inputStream ->
  textContent = IOUtils.toString(inputStream, StandardCharsets.UTF_8);
  // Do something with text here
```

<table>
<tr><td></td><td>

```
} as InputStreamCallback)
//get the different dates
def arrAttrs=textContent.split('","');
def elements;
def tagDate="data";
def date;
def dateTop=Date.parse("yyyy-MM-dd",flowFile.getAttribute("dateStart"));
def pattern = "yyyy-MM-ddHH:mm:ss.sss"
//split the entire json
for (attrs in arrAttrs)
        {
        //split the key value
        elements=attrs.split('":"');
        //we only need the date
        if (tagDate==elements[0])
        {
        //println(elements[1])
        date =  Date.parse(pattern,elements[1].replace("T",""));
        println (date)
        if (dateTop==null)
        {
        dateTop=date;
        }
        else
        {
        if (dateTop<date)
        {
        dateTop=date;
        }
        }

        }
        }
result=dateTop.format("yyyy-MM-dd");
//replacing the output content
flowFile = session.write(flowFile, {outputStream ->
  outputStream.write(result.getBytes(StandardCharsets.UTF_8))
} as OutputStreamCallback)

//redirect the flowfile as appropiate
session.transfer(flowFile, REL_SUCCESS)
```

</td></tr>
<tr><td>RouteOnContent</td><td>

Property	Value
Match Requirement	**content must match exactly**
Character Set	**UTF-8**
Content Buffer Size	**1 MB**
dateEqual	${dateStart}

</td></tr>
<tr><td>setFileName</td><td>

Property	Value
Delete Attributes Expression	No value set
Store State	**Do not store state**
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	**100**
filename	${municipi}_${pollutant}

</td></tr>
</table>

insertIndexCItyHDFS

Property	Value
Hadoop Configuration Resources	#{HAD_CONFIG_FILES_PATH}
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
Directory	**#{HAD_PATH_INDEX_DATECITY}**
Conflict Resolution Strategy	**replace**
Writing Strategy	**Simple write**
Block Size	No value set

Proseguimos con la línea principal de proceso y finalización

Script split hours json

Property	Value
Script Engine	**Groovy**
Script File	No value set
Script Body	import org.apache.commons.io.IOUtils;...
Module Directory	No value set

Código Groovy:

```
import org.apache.commons.io.IOUtils;
import java.nio.charset.StandardCharsets;
//get flowfile, if not, abort
flowFile = session.get();
if(!flowFile)return;
def textContent = '';
def timeStampSecond = flowFile.getAttribute("timeStampSecond");
def separator = ",";

// Cast a closure with an inputStream parameter to InputStreamCallback
session.read(flowFile, {inputStream ->
  textContent = IOUtils.toString(inputStream, StandardCharsets.UTF_8);
  // Do something with text here

} as InputStreamCallback)
//format the contentText
def arrAtts=textContent.replace("{","").replace("}","").split (",");//replace the {} from begin and end of the json string
def hourTags=["h01","h02","h03","h04","h05","h06","h07","h08","h09","h10","h11","h12","h13","h14","h15","h16","h17","h18","h19","h20","h21","h22","h23","h24"];//the hour list accepted tags
def bodyTags=["codi_eoi","nom_estacio","data","magnitud","contaminant","unitats","codi_comarca","nom_comarca","longitud","latitud"];
String body="";
String result="";
def hora;
def horaTag;
def quantitat;
def element;
//creating the body data, alter we will the hour
for (par in arrAtts)
   {
            for (tag in bodyTags)
             {
            if (par.contains(tag))
```

```
                    {
                    //correcting the body date
                    if (tag.equals('data'))
                            {
                            par=par.replaceAll('T',' ');//the date comes
yyyy-MM-ddTHH:mm:ss
                            }
                    if (body=="")body=par;
                    else body=body+","+par;
            }
            }
            }
//now we will traverse the hours (hxx format)
for (par in arrAtts)
            {
            for (tag in hourTags)
            {
             if (par.contains(tag))
                            {
            horaTag=par.split(":")[0];//we get the key (in hxx format)
            quantitat=par.split(":")[1];//we get the ammount
            horaTag=horaTag.replaceAll('"','');//remove the "
            horaTag='"hora":"'+horaTag.replace("h","").toInteger()+'"'; //we want the time
as integer not like hxx
                    //create the new element
                    element=body+',"quantitat":'+quantitat;
                    element+=','+horaTag;
                    element+=',"timeStampSecond":"'+timeStampSecond+'"';
                    if (result!="")  result+=separator; //add the element separator if
needed
                    result+="{"+element+"}";
            }
            }
   }
result="["+result+"]";//we store it as array
//replacing the output content
flowFile = session.write(flowFile, {outputStream ->
  outputStream.write(result.getBytes(StandardCharsets.UTF_8))
} as OutputStreamCallback)

//redirect the flowfile as appropiate
session.transfer(flowFile, REL_SUCCESS)
```

SplitByHour

Property	Value
JsonPath Expression	$.*
Null Value Representation	empty string
Max String Length	20 MB

getRowVariables

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
Max String Length	20 MB
codEoi	$.["codi_eoi"]
contaminant	$.["contaminant"]
date	$.["data"]
hora	$.['hora']

setFileName

Property	Value
Delete Attributes Expression	No value set
Store State	**Do not store state**
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	**100**
filename	${codEoi}_${contaminant}_${date:replaceAll("T"," "):toDate...

filename:
${codEoi}_${contaminant}_${date:replaceAll("T"," "):toDate("yyyy-MM-dd HH:mm:ss.SSS"):format("yyyyMMdd_HHmm")}_${hora}

insertProcessedHDFS

Property	Value
Hadoop Configuration Resources	#{HAD_CONFIG_FILES_PATH}
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
Directory	**#{HAD_PATH_PROCESSED_POLLUTANT}**
Conflict Resolution Strategy	**replace**
Writing Strategy	**Simple write**
Block Size	No value set

Sección Hadoop

(mostraremos la estructura de carpetas creada)

```
# hdfs dfs -ls -R /TFM |grep "^d"
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:46 /TFM/Index
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:53 /TFM/Index/Date
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:54 /TFM/Index/Date/City
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:56 /TFM/Index/Date/Station
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:18 /TFM/NifiProcessed
drwxr-xr-x   - root supergroup          0 2024-05-22 20:13 /TFM/NifiProcessed/Meteo
drwxr-xr-x   - nifi supergroup          0 2024-05-22 20:13 /TFM/NifiProcessed/Pollutant
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:46 /TFM/Original
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:54 /TFM/Original/Meteo
drwxr-xr-x   - nifi supergroup          0 2024-06-14 17:54 /TFM/Original/Pollutant
drwxr-xr-x   - root supergroup          0 2024-05-24 18:06 /TFM/PostProcessed
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/AggDay
drwxr-xr-x   - root supergroup          0 2024-06-11 21:26 /TFM/PostProcessed/AggMeteo
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/AggYear
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/Contaminants
drwxr-xr-x   - root supergroup          0 2024-05-28 19:58 /TFM/PostProcessed/Municipis
drwxr-xr-x   - root supergroup          0 2024-05-29 19:58 /TFM/PostProcessed/coreData
drwxr-xr-x   - root supergroup          0 2024-05-24 18:06 /TFM/PostProcessed/relacionMeteo
drwxr-xr-x   - nifi supergroup          0 2024-05-19 09:57 /TFM/Predictions
drwxr-xr-x   - root supergroup          0 2024-05-19 09:18 /TFM/Processed
drwxr-xr-x   - root supergroup          0 2024-05-27 18:22 /TFM/Processed/Meteo
drwxr-xr-x   - root supergroup          0 2024-05-22 20:12 /TFM/Processed/Pollutant
drwxr-xr-x   - root supergroup          0 2024-05-27 18:51 /TFM/Sources
drwxr-xr-x   - root supergroup          0 2024-05-20 10:22 /TFM/Sources/ObjetivosCalidad
drwxr-xr-x   - root supergroup          0 2024-05-27 18:21 /TFM/Sources/historicMeteo
drwxr-xr-x   - root supergroup          0 2024-05-19 09:04 /TFM/Sources/relacionFestivos
drwxr-xr-x   - root supergroup          0 2024-05-24 18:10 /TFM/Sources/relacionMeteo
```

drwxr-xr-x - root supergroup 0 2024-05-27 18:49 /TFM/Sources/relacionMeteoMetadata

Sección HIVE

(scripts de creación de tablas, queries de consulta y filtrado)

Scripts de creación de tablas	
relacio_estacions	create external table if not exists relacio_estacions(municipi string, codi_eoi string, codi_estacio string, notes string) comment 'relacion de estaciones meteorologicas y municipios' row format delimited fields terminated by '\t' stored as textfile location '/TFM/Sources/relacionMeteo/';
relacio_festius	create external table if not exists relacio_festius(data_festiva date) comment 'relacion de fechas festivas aplicables' row format delimited fields terminated by '\t' stored as textfile location '/TFM/Sources/relacionFestivos/';
nifi_polucio	CREATE EXTERNAL TABLE IF NOT EXISTS nifi_polucio (codi_eoi string,nom_estacio string,data string,magnitud string,contaminant string,unitats string,codi_comarca string,latitud string,longitud string,quantitat string,hora string,timeStampSecond string) COMMENT 'data preprocessed by Nifi about pollution' ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE LOCATION '/TFM/NifiProcessed/Pollutant/';
processed_polucio	CREATE EXTERNAL TABLE IF NOT EXISTS processed_polucio (codi_eoi string,nom_estacio string,data string,magnitud string,contaminant string,unitats string,codi_comarca string,latitud string,longitud string,quantitat string,hora string,timeStampSecond string) COMMENT 'storage of processed pollution data' ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE LOCATION '/TFM/Processed/Pollutant/';
nifi_meteo	CREATE EXTERNAL TABLE IF NOT EXISTS nifi_meteo (id string,codi_estacio string,codi_variable string,data_lectura string,valor_lectura string,codi_base string,timeStampSecond string) COMMENT 'data preprocessed by Nifi about meteorological data' ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE LOCATION '/TFM/NifiProcessed/Meteo/';
processed_meteo	CREATE EXTERNAL TABLE IF NOT EXISTS processed_meteo (id string,codi_estacio string,codi_variable string,data_lectura string,valor_lectura string,codi_base string,timeStampSecond string) COMMENT 'storage of processed meteorological data' ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE LOCATION '/TFM/Processed/Meteo/';
postProcessed_data	CREATE EXTERNAL TABLE IF NOT EXISTS postProcessed_data (codi_eoi string,nom_estacio string,data string, magnitud int,contaminant string,unitats string, codi_comarca string,latitud string,longitud string, quantitat double,hora string,codi_estacio string,codi_variable string, descripcio_variable string, data_lectura string, valor_lectura double,codi_base string, festiuTag int) COMMENT 'merged data after hive queries' ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

	STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/coreData/';
temp_historic_meteo	CREATE EXTERNAL TABLE IF NOT EXISTS temp_historic_meteo (id string,codi_estacio string,codi_variable string,data_lectura string,empty string,valor_lectura string,codiV string, codi_base string,timeStampSecond string) COMMENT 'data downloaded from XEMA in csv ; format' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/Sources/historicMeteo/';
objectius_qualitat	CREATE EXTERNAL TABLE IF NOT EXISTS objectius_qualitat (idContaminant string,periode string,quantitat double,origen string) COMMENT 'Objectius contaminants per BOE i WHO in csv tab format' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE LOCATION '/TFM/Sources/ObjetivosCalidad/';
agg_polucio_day	CREATE EXTERNAL TABLE IF NOT EXISTS agg_polucio_day (codi_comarca string,nom_estacio string,codi_eoi string,contaminant string, magnitud string, data date,latitud string,longitud string,festiuTag int,limitDia double, limitAny double,contaminant_excedit int, valor double) COMMENT 'agregat de contaminació diaria' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/AggDay/';
agg_polucio_year	CREATE EXTERNAL TABLE IF NOT EXISTS agg_polucio_year (codi_comarca string,nom_estacio string,codi_eoi string,contaminant string, magnitud string, any string,latitud string,longitud string,limitDia double, limitAny double,contaminant_excedit int, valor double) COMMENT 'agregat de contaminació year' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/AggYear/';
agg_polucio_meteo	CREATE EXTERNAL TABLE IF NOT EXISTS agg_polucio_meteo (codi_comarca string, nom_estacio string, codi_eoi string, contaminant string, data_lectura string, hora string, codi_estacio string, codi_variable string, descripcio_variable string, valor_lectura double, festiuTag int, latitud string, longitud string, limitDiari double, limitAnual double, quantitat double, unitat_polucio string, excedit_diari string, excedit_horari int, excedit_any int) COMMENT 'agregat de contaminació per dia i contaminant' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/AggMeteo/';
agg_polucio_meteo_complete	CREATE EXTERNAL TABLE IF NOT EXISTS agg_polucio_meteo_complete (codi_comarca string, nom_estacio string, codi_eoi string, contaminant string, data_lectura string, hora string, codi_estacio string, codi_variable string, descripcio_variable string, valor_lectura double, festiuTag int, latitud string, longitud string, limitDiari double, limitAnual double, quantitat double, unitat_polucio string, excedit_diari string, excedit_horari int, excedit_any int) COMMENT 'agregat de contaminació per dia i contaminant' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/AggMeteoComplete/';
municipis	CREATE EXTERNAL TABLE IF NOT EXISTS municipis (nom_municipi string, codi_eoi string)

	COMMENT 'Tabla utilitzada per crear filtres comuns per a explotació posterior' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/Municipis/';
contaminants	CREATE EXTERNAL TABLE IF NOT EXISTS contaminants (contaminant string) COMMENT 'Tabla utilitzada per crear filtres comuns per a explotació posterior' ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' STORED AS TEXTFILE LOCATION '/TFM/PostProcessed/Contaminants/';
relacio_metadata_meteo	CREATE EXTERNAL TABLE IF NOT EXISTS relacio_metadata_meteo (codi_variable string, descripcio_variable string) COMMENT 'Tabla utilitzada per agregar la descripció del metadata de meterologia' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE LOCATION '/TFM/Sources/relacionMeteoMetadata/';

Scripts de consultas y filtrado en HUE	
agg polución meteo	INSERT OVERWRITE TABLE agg_polucio_meteo Select ppd.codi_comarca, ppd.nom_estacio, ppd.codi_eoi, ppd.contaminant, ppd.data_lectura, ppd.hora, ppd.codi_estacio, ppd.codi_variable, ppd.descripcio_variable, ppd.valor_lectura, ppd.festiuTag, ppd.latitud, ppd.longitud, CASE WHEN oqd.quantitat IS NOT NULL THEN oqd.quantitat ELSE 0 END as limitDiari, CASE WHEN oqa.quantitat IS NOT NULL THEN oqa.quantitat ELSE 0 END as limitAnual, AVG(ppd.quantitat) as quantitat, ppd.unitats, CASE WHEN agd.limitdia IS NOT NULL THEN from_unixtime(unix_timestamp(agd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") ELSE 0 END as exceditDiari, CASE WHEN agd.limitdia IS NOT NULL THEN cast(agd.limitdia<AVG(ppd.quantitat) as int) ELSE 0 END as exceditHorari, CASE WHEN agy.any IS NOT NULL THEN cast(agy.any as int) ELSE 0 END as exceditAny FROM postprocessed_data ppd LEFT JOIN objectius_qualitat oqd ON (oqd.idcontaminant=ppd.contaminant AND oqd.origen='WHO' AND oqd.periode='d') LEFT JOIN objectius_qualitat oqa ON (oqa.idcontaminant=ppd.contaminant AND oqa.origen='WHO' AND oqa.periode='a') LEFT JOIN agg_polucio_day agd ON (agd.contaminant_excedit=1 AND ppd.codi_eoi=agd.codi_eoi AND ppd.contaminant=agd.contaminant AND agd.data=cast(from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") as date)) LEFT JOIN agg_polucio_year agy ON (agy.contaminant_excedit=1 AND ppd.codi_eoi=agy.codi_eoi AND ppd.contaminant=agy.contaminant AND agy.any=from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy")) WHERE from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd

	HH:mm:ss'),"yyyy")>=(from_unixtime(unix_timestamp(),"yyyy")-2) GROUP BY ppd.codi_comarca,ppd.nom_estacio,ppd.codi_eoi,ppd.contaminant,ppd.data_lectura,ppd.hora,ppd.codi_estacio,ppd.codi_variable,ppd.descripcio_variable,ppd.valor_lectura,ppd.festiuTag,ppd.latitud,ppd.longitud,oqd.quantitat,oqa.quantitat,ppd.unitats, agd.limitdia,agd.data,agy.any;
agg polució meteo historic	INSERT OVERWRITE TABLE agg_polucio_meteo_complete Select ppd.codi_comarca, ppd.nom_estacio, ppd.codi_eoi, ppd.contaminant, ppd.data_lectura, ppd.hora, ppd.codi_estacio, ppd.codi_variable, ppd.descripcio_variable, ppd.valor_lectura, ppd.festiuTag, ppd.latitud, ppd.longitud, CASE WHEN oqd.quantitat IS NOT NULL THEN oqd.quantitat ELSE 0 END as limitDiari, CASE WHEN oqa.quantitat IS NOT NULL THEN oqa.quantitat ELSE 0 END as limitAnual, AVG(ppd.quantitat) as quantitat, ppd.unitats, CASE WHEN agd.limitdia IS NOT NULL THEN from_unixtime(unix_timestamp(agd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") ELSE 0 END as exceditDiari, CASE WHEN agd.limitdia IS NOT NULL THEN cast(agd.limitdia<AVG(ppd.quantitat) as int) ELSE 0 END as exceditHorari, CASE WHEN agy.any IS NOT NULL THEN cast(agy.any as int) ELSE 0 END as exceditAny FROM postprocessed_data ppd LEFT JOIN objectius_qualitat oqd ON (oqd.idcontaminant=ppd.contaminant AND oqd.origen='WHO' AND oqd.periode='d') LEFT JOIN objectius_qualitat oqa ON (oqa.idcontaminant=ppd.contaminant AND oqa.origen='WHO' AND oqa.periode='a') LEFT JOIN agg_polucio_day agd ON (agd.contaminant_excedit=1 AND ppd.codi_eoi=agd.codi_eoi AND ppd.contaminant=agd.contaminant AND agd.data=cast(from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy-MM-dd") as date)) LEFT JOIN agg_polucio_year agy ON (agy.contaminant_excedit=1 AND ppd.codi_eoi=agy.codi_eoi AND ppd.contaminant=agy.contaminant AND agy.any=from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd HH:mm:ss'),"yyyy")) GROUP BY ppd.codi_comarca,ppd.nom_estacio,ppd.codi_eoi,ppd.contaminant,ppd.data_lectura,ppd.hora,ppd.codi_estacio,ppd.codi_variable,ppd.descripcio_variable,ppd.valor_lectura,ppd.festiuTag,ppd.latitud,ppd.longitud,oqd.quantitat,oqa.quantitat,ppd.unitats, agd.limitdia,agd.data,agy.any;
create PostProcessedData	INSERT OVERWRITE TABLE postProcessed_data SELECT pp.codi_eoi, pp.nom_estacio, from_unixtime(unix_timestamp(pp.data, 'yyyy-MM-dd HH:mm:ss.SSS'),"yyyy-MM-dd HH:mm:ss") data, cast(pp.magnitud as int) magnitud, pp.contaminant, pp.unitats, pp.codi_comarca, pp.latitud, pp.longitud, cast(pp.quantitat as double) quantitat, format_number(cast(pp.hora as int),"00") hora, re.codi_estacio, pm.codi_variable,

	rmm.descripcio_variable, from_unixtime(unix_timestamp(pm.data_lectura, 'dd/MM/yyyy HH:mm:'),"yyyy-MM-dd HH:mm:ss") data_lectura, cast(pm.valor_lectura as double) valor_lectura, pm.codi_base, CASE WHEN rf.data_festiva IS NOT null OR from_unixtime(unix_timestamp(pm.data_lectura, 'dd/MM/yyyy HH:mm'),"u")=7 then 1 else 0 end festiuTag FROM processed_polucio pp INNER JOIN relacio_estacions re ON (re.codi_eoi =pp.codi_eoi) INNER JOIN processed_meteo pm ON (pm.codi_estacio=re.codi_estacio AND from_unixtime(unix_timestamp(pm.data_lectura, 'dd/MM/yyyy HH:mm'),"yyyy-MM-dd HH:mm")=(from_unixtime(unix_timestamp(pp.data, 'yyyy-MM-dd HH:mm:ss.SSS'),"yyyy-MM-dd")|| ' ' ||format_number(cast(pp.hora as int),'00')||':00')) LEFT JOIN relacio_festius rf ON (rf.data_festiva=from_unixtime(unix_timestamp(pp.data, 'yyyy-MM-dd HH:mm:ss.SSS'),"yyyy-MM-dd")) LEFT JOIN relacio_metadata_meteo rmm ON (rmm.codi_variable=pm.codi_variable);
filter contaminant table	INSERT OVERWRITE TABLE contaminants select distinct contaminant from postprocessed_data;
filter municipi table	INSERT OVERWRITE TABLE municipis select distinct nom_estacio,codi_eoi from postprocessed_data;
transfer nifi meteo to processed	INSERT INTO TABLE processed_meteo SELECT * FROM nifi_meteo;
transfer nifi polució to processed	INSERT INTO TABLE processed_polucio SELECT * FROM nifi_polucio;
clear nifi polucio	INSERT OVERWRITE TABLE nifi_polucio SELECT * FROM nifi_polucio WHERE 1=2;
clear nifi meteo	INSERT OVERWRITE TABLE nifi_meteo SELECT * FROM nifi_meteo WHERE 1=2;
purge processed meteo	INSERT OVERWRITE TABLE processed_meteo select distinct * from processed_meteo;
purge processed polucio	INSERT OVERWRITE TABLE processed_polucio select distinct * from processed_polucio;
agg polución by day	INSERT OVERWRITE TABLE agg_polucio_day Select ppd.codi_comarca,ppd.nom_estacio,ppd.codi_eoi,ppd.contaminant,ppd.unitats, ppd.data,ppd.latitud,ppd.longitud,ppd.festiuTag, CASE WHEN oqd.quantitat IS NOT NULL THEN oqd.quantitat ELSE 0 END limitDia, CASE WHEN oqa.quantitat IS NOT NULL THEN oqa.quantitat ELSE 0 END limitYear,CASE WHEN oqd.quantitat IS NOT NULL THEN CASE WHEN AVG(ppd.quantitat)>oqd.quantitat THEN 1 ELSE 0 END ELSE 0 END contaminant_exceeded, AVG(ppd.quantitat) valor FROM postProcessed_data ppd LEFT JOIN objectius_qualitat oqd ON (oqd.idcontaminant=ppd.contaminant AND oqd.origen='WHO' AND oqd.periode='d')

	LEFT JOIN objectius_qualitat oqa ON (oqa.idcontaminant=ppd.contaminant AND oqa.origen='WHO' AND oqa.periode='a') GROUP BY codi_comarca,nom_estacio,codi_eoi,contaminant,unitats, data,ppd.latitud,ppd.longitud,ppd.festiuTag, oqd.quantitat,oqa.quantitat;
agg polucio by year	INSERT OVERWRITE TABLE agg_polucio_year Select ppd.codi_comarca,ppd.nom_estacio,ppd.codi_eoi,ppd.contaminant,ppd.unitats, from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd'),"yyyy") any,ppd.latitud,ppd.longitud,CASE WHEN oqd.quantitat IS NOT NULL THEN oqd.quantitat ELSE 0 END limitDia, CASE WHEN oqa.quantitat IS NOT NULL THEN oqa.quantitat ELSE 0 END limitYear,CASE WHEN oqa.quantitat IS NOT NULL THEN CASE WHEN AVG(ppd.quantitat)>oqa.quantitat THEN 1 ELSE 0 END ELSE 0 END contaminant_exceeded, AVG(ppd.quantitat) valor FROM postprocessed_data ppd LEFT JOIN objectius_qualitat oqd ON (oqd.idcontaminant=ppd.contaminant AND oqd.origen='WHO' AND oqd.periode='d') LEFT JOIN objectius_qualitat oqa ON (oqa.idcontaminant=ppd.contaminant AND oqa.origen='WHO' AND oqa.periode='a') GROUP BY codi_comarca,nom_estacio,codi_eoi,contaminant,unitats, from_unixtime(unix_timestamp(ppd.data, 'yyyy-MM-dd'),"yyyy"),ppd.latitud,ppd.longitud,oqd.quantitat,oqa.quantitat;

Sección Power Bi

Fórmula DAX de creación de tabla CalendarDateTime con saltos cada hora	let
	FromDateTime = #datetime(2022,1,1,0,0,0),
	ToDateTime = #datetime(2024,12,31,23,59,0),
	ListofDataTime = List.DateTimes(FromDateTime,Duration.Days(ToDateTime-FromDateTime)*24,#duration(0,1,0,0)),
	#"Convertida en tabla" = Table.FromList(ListofDataTime, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
	#"Columnas con nombre cambiado" = Table.RenameColumns(#"Convertida en tabla",{{"Column1", "DateTime"}}),
	#"Tipo cambiado" = Table.TransformColumnTypes(#"Columnas con nombre cambiado",{{"DateTime", type datetime}}),
	#"Personalizada agregada" = Table.AddColumn(#"Tipo cambiado", "Any", each Date.Year([DateTime])),
	#"Tipo cambiado1" = Table.TransformColumnTypes(#"Personalizada agregada",{{"Any", Int64.Type}}),
	#"Personalizada agregada1" = Table.AddColumn(#"Tipo cambiado1", "Personalizado", each Date.Month([DateTime])),
	#"Columnas con nombre cambiado1" = Table.RenameColumns(#"Personalizada agregada1",{{"Personalizado", "Mes"}}),
	#"Tipo cambiado2" = Table.TransformColumnTypes(#"Columnas con nombre cambiado1",{{"Mes", Int64.Type}}),
	#"Personalizada agregada2" = Table.AddColumn(#"Tipo cambiado2", "Día", each Date.Day([DateTime])),
	#"Tipo cambiado3" = Table.TransformColumnTypes(#"Personalizada agregada2",{{"Día", Int64.Type}}),
	#"Personalizada agregada3" = Table.AddColumn(#"Tipo cambiado3", "Dia de la setmana", each Date.DayOfWeek([DateTime])),
	#"Tipo cambiado4" = Table.TransformColumnTypes(#"Personalizada agregada3",{{"Dia de la setmana", Int64.Type}}),
	#"Personalizada agregada4" = Table.AddColumn(#"Tipo cambiado4", "Día Text", each Date.DayOfWeekName([DateTime])),
	#"Personalizada agregada5" = Table.AddColumn(#"Personalizada agregada4", "Mes Text", each Date.MonthName([DateTime])),
	#"Tipo cambiado5" = Table.TransformColumnTypes(#"Personalizada agregada5",{{"Día Text", type text}, {"Mes Text", type text}}),
	#"Poner En Mayúsculas Cada Palabra" = Table.TransformColumns(#"Tipo cambiado5",{{"Día Text", Text.Proper, type text}, {"Mes Text", Text.Proper, type text}}),

	#"Personalizada agregada6" = Table.AddColumn(#"Poner En Mayúsculas Cada Palabra", "Trimestre", each Date.QuarterOfYear([DateTime])), #"Columnas reordenadas" = Table.ReorderColumns(#"Personalizada agregada6",{"DateTime", "Any", "Trimestre", "Mes", "Día", "Dia de la setmana", "Día Text", "Mes Text"}), #"Personalizada agregada7" = Table.AddColumn(#"Columnas reordenadas", "Hora", each Time.Hour([DateTime])), #"Tipo cambiado6" = Table.TransformColumnTypes(#"Personalizada agregada7",{{"Hora", Int64.Type}}), #"Personalizada agregada8" = Table.AddColumn(#"Tipo cambiado6", "Setmana de l'any", each Date.WeekOfYear([DateTime])), #"Personalizada agregada9" = Table.AddColumn(#"Personalizada agregada8", "Time", each DateTime.Time([DateTime])), #"Personalizada agregada10" = Table.AddColumn(#"Personalizada agregada9", "Date", each DateTime.Date([DateTime])), #"Tipo cambiado7" = Table.TransformColumnTypes(#"Personalizada agregada10",{{"Time", type time}, {"Date", type date}, {"Trimestre", Int64.Type}}), #"Personalizada agregada11" = Table.AddColumn(#"Tipo cambiado7", "Personalizado", each DateTime.ToText([DateTime], [Format="yyyyMMddHH"])) in #"Personalizada agregada11"
Fórmula dax para discretización de variables climatológicas	valor lectura discretizada = var discret=agg_polucio_meteo[valor_lectura] var result=switch(TRUE(), agg_polucio_meteo[descripcio_variable]="Temperatura",round(agg_polucio_meteo[valor_lectura],0) , agg_polucio_meteo[descripcio_variable]="Vel. vent (a 1m)",FORMAT(round(agg_polucio_meteo[valor_lectura],0),"0"), agg_polucio_meteo[descripcio_variable]="Humitat",round(agg_polucio_meteo[valor_lectura]/5,0)*5 , agg_polucio_meteo[descripcio_variable]="Direcció del vent (a 1m)",round(agg_polucio_meteo[valor_lectura]/20,0)*20, agg_polucio_meteo[descripcio_variable]="Pressió atmosferica",round(agg_polucio_meteo[valor_lectura]/3,0)*3, discret) return value(result)