



Universidad de Valladolid

Escuela de Ingeniería Informática Segovia

Trabajo de Fin de Grado

Grado en Ingeniería Informática de Servicios y Aplicaciones

Sistema modular de almacenamiento y
monitorización de la calidad del aire en
Castilla y León

Autor:

Javier Herrero Ballesteros

Tutor:

Anibal Bregon Bregon



Universidad de Valladolid

Escuela de Ingeniería Informática Segovia

Trabajo de Fin de Grado

Grado en Ingeniería Informática de Servicios y Aplicaciones

Sistema modular de almacenamiento y
monitorización de la calidad del aire en
Castilla y León

Autor:

Javier Herrero Ballesteros

Índice General

1.	Introducción	12
1.1	Contexto y motivación.....	12
1.2	Objetivos	14
1.3	Alcance del proyecto.....	15
1.4	Distribución del contenido.....	15
2.	Marco teórico	17
2.1	Tecnologías empleadas	17
2.1.1	Docker.....	17
2.1.2	Apache Nifi.....	19
2.1.3	MinIO.....	21
2.1.4	InfluxDB	23
2.1.5	Grafana.....	25
2.2	Control de la calidad del aire	27
2.2.1	Elementos contaminantes.....	27
2.2.2	Influencia de las condiciones meteorológicas.....	31
2.2.3	Captura de datos.....	32
3.	Arquitectura del sistema	34
3.1	Introducción a la arquitectura	34

3.1.1	Arquitectura dockerizada	35
3.2	Comunicación entre servicios	36
3.2.1	Red interna de Docker	36
3.2.2	Flujos de comunicación entre servicios	39
4.	Análisis y diseño	42
4.1	Conocimiento de los datos	42
4.1.1.	Fuentes de información	42
4.1.2	Estaciones de control	43
4.1.3	Formatos y estructura de datos	44
4.1.4	Umbrales de calidad del aire	52
4.2	Modelado de dominio	53
4.3	Diseño del flujo de datos	58
5.	ETL	61
5.1	Extracción	64
5.1.1.	Origen	64
5.1.2.	Ingesta	66
5.1.3.	Almacenamiento	69
5.2	Transformación	71
5.2.1.	Procesamiento	71
5.2.1.	Persistencia	77

5.3 Carga	78
5.3.1. Transferencia de datos	78
5.3.2. Consumo de datos	80
6. Exploración	82
6.1 Planificación	82
6.1.1. Propósito	82
6.1.2. Enfoque	83
6.1.3. Diseño	83
6.2 Implementación.....	84
6.2.1. Construcción de las visualizaciones.....	84
6.2.2. Detalle de las visualizaciones	86
6.2.3. Interacción y navegación	110
7. Conclusiones y trabajos futuros	114
7.1 Trabajos futuros	115
Bibliografía	116

Índice de figuras

Figura 1.1. Mapa contaminación del aire a nivel global [1].	13
Figura 2.1. Comparativa de Docker y VM. Fuente: Elaboración propia	18
Figura 2.2. Comando en PowerShell para levantar el docker-compose. Fuente: Elaboración propia	18
Figura 2.3. Esquema de estructura utilizada para Docker Compose. Fuente: Elaboración propia	19
Figura 2.4. Funcionamiento flujo de datos Apache Nifi. Fuente: Elaboración propia	20
Figura 2.5. Organización de los objetos en MinIO. Fuente: Elaboración propia.	21
Figura 2.6. Formato Line Protocol. Fuente: Elaboración propia.	24
Figura 2.7. Tipos de panel de visualización Grafana. Fuente: Interfaz de control Grafana.	26
Figura 2.8. Mapa global de concentración NO2 [17].	29
Figura 2.9. Mapa global de material particulado 2.5 [21]	31
Figura 2.10. Distribución de las estaciones de control de calidad del aire en España [23].	33
Figura 3.1. Diagrama del flujo de información entre los servicios. Fuente: Elaboración propia .	35
Figura 3.2. Arquitectura Docker [30].	36
Figura 3.3. Archivo docker-compose.yml. Fuente: Elaboración propia.....	38
Figura 3.4. Captura con los servicios ejecutados desde Docker Desktop. Fuente: Elaboración propia	40
Figura 3.5. Flujo de comunicación entre servicios. Fuente: Elaboración propia.....	41
Figura 4.1. Diagrama fases en la corriente de datos. Fuente: Elaboración propia.....	44

Figura 4.2. Procesos para estructura y formateo de datos. Fuente: Elaboración propia	45
Figura 4.3. Ejemplo JSON fase 1A. Fuente: Elaboración propia	46
Figura 4.4. Ejemplo JSON en fase 2A. Fuente: Elaboración propia	47
Figura 4.5. Ejemplo LineProtocol para una provincia. Fuente: Elaboración propia	48
Figura 4.6. Ejemplo JSON en fase 1B. Fuente: Elaboración propia.....	49
Figura 4.7. Ejemplo JSON en fase 2B. Fuente: Elaboración propia.....	50
Figura 4.8. Ejemplo de datos en .lp en la fase 3B. Fuente: Elaboración propia	51
Figura 4.9. Diagrama E-R de datos en crudo. Fuente: Elaboración propia	54
Figura 4.10. Diagrama E-R de datos procesados. Fuente: Elaboración propia	55
Figura 4.11. Esquema de diseño del Data-Flow. Fuente: Elaboración propia.....	59
Figura 5.1. Diagrama de flujo proceso ETL datos meteorológicos. Fuente: Elaboración propia.	62
Figura 5.2. Diagrama de flujo proceso ETL datos de calidad del aire. Fuente: Elaboración propia	63
Figura 5.3. Esquema corriente de datos oficiales. Fuente: Elaboración propia.....	65
Figura 5.4. Respuesta JSON de la API devolviendo URL con los datos solicitados []......	67
Figura 5.5. Buckets disponibles en MinIO. Fuente: Elaboración propia.....	69
Figura 5.6. Propiedades del procesador PutS3Object en el flujo de meteorología. Fuente:	
Elaboración propia	69
Figura 5.7. Propiedades del procesador PutS3Object para calidad del aire. Fuente:	
Elaboración propia	70

Figura 5.8. Consulta correspondiente al QueryRecord para la meteorología.	Fuente:
Elaboración propia	72
Figura 5.9. Propiedades del procesador QueryRecord para calidad del aire.	Fuente:
Elaboración propia	74
Figura 5.10. Script utilizado en el procesador B10.	Fuente: Elaboración propia
	75
Figura 5.11. Propiedades del procesador PutS3Object.	Fuente: Elaboración propia
	76
Figura 5.12. Propiedades del procesador PutS3Object para la meteorología.	Fuente:
Elaboración propia	77
Figura 5.13. Propiedades del procesador PutS3Object para la meteorología.	Fuente:
Elaboración propia	77
Figura 5.14. Diagrama explicativo del proceso de transferencia hacia InfluxDB.	
Fuente: Elaboración propia	79
Figura 5.15. Esquema de interacciones Grafana-InfluxDB.	Fuente: Elaboración propia
	81
Figura 6.1. Grafana, apartados desplegados del dashboard.	Fuente: Elaboración propia.....
	85
Figura 6.2. Grafana, apartado temperaturas.	Fuente: Elaboración propia
	87
Figura 6.3. Grafana, apartado NO2.	Fuente: Elaboración propia
	89
Figura 6.4. Grafana, apartado NO.	Fuente: Elaboración propia
	92
Figura 6.5. Grafana, apartado SO2.	Fuente: Elaboración propia.....
	95
Figura 6.6. Grafana, apartado CO.	Fuente: Elaboración propia
	97
Figura 6.7. Grafana, apartado PM2.5.	Fuente: Elaboración propia
	100
Figura 6.8. Grafana, apartado PM10.	Fuente: Elaboración propia
	103
Figura 6.9. Grafana, apartado O3.	Fuente: Elaboración propia
	106

Figura 6.10. Grafana, apartado matriz de contaminantes-temperatura. Fuente: Elaboración propia
109

Figura 6.11. Filtro de provincias del dashboard Grafana. Fuente: Elaboración propia 110

Figura 6.12. Filtro para rango de tiempo en dashboard Grafana. Fuente: Elaboración propia... 111

Figura 6.13. Grafana, solo valores O3 visibles. Fuente: Elaboración propia 112

Figura 6.14. Grafana, valores O3 + valor umbral visibles. Fuente: Elaboración propia 112

Figura 6.15. Grafana, valores O3 + umbral + promedio móvil. Fuente: Elaboración propia..... 113

Índice de tablas

Tabla 4.1. Cuadro con los umbrales de los contaminantes. Fuente: Elaboración propia	53
Tabla 4.2. Diccionario de EstacionAire. Fuente: Elaboración propia	56
Tabla 4.3. Diccionario de EstacionMeteo. Fuente: Elaboración propia	56
Tabla 4.4. Diccionario de RegistroAire. Fuente: Elaboración propia.....	57
Tabla 4.5. Diccionario de RegistroMeteo. Fuente: Elaboración propia	58
Tabla 6.1. Panel, valor máximo de temperaturas. Fuente: Elaboración propia	87
Tabla 6.2. Panel, valor mínimo de temperaturas. Fuente: Elaboración propia	88
Tabla 6.3. Panel, promedio de temperaturas máximas. Fuente: Elaboración propia.....	88
Tabla 6.4. Panel, promedio de temperaturas mínimas. Fuente: Elaboración propia	88
Tabla 6.5. Panel, valor máximo NO2. Fuente: Elaboración propia	90
Tabla 6.6. Panel, valor mínimo NO2. Fuente: Elaboración propia.....	90
Tabla 6.7. Panel, valor promedio NO2s. Fuente: Elaboración propia	91
Tabla 6.8. Panel, gráfico valores NO2. Fuente: Elaboración propia	91
Tabla 6.9. Panel, valor máximo NO. Fuente: Elaboración propia	92
Tabla 6.10. Panel, valor mínimo NO. Fuente: Elaboración propia.....	93
Tabla 6.11. Panel, valor promedio NO. Fuente: Elaboración propia.....	93
Tabla 6.12. Panel, gráfico con valores NO. Fuente: Elaboración propia.....	94
Tabla 6.13. Panel, valor máximo SO2. Fuente: Elaboración propia.....	95

Tabla 6.14. Panel, valor mínimo SO ₂ . Fuente: Elaboración propia	96
Tabla 6.15. Panel, valor promedio de SO ₂ . Fuente: Elaboración propia.....	96
Tabla 6.16. Panel, gráfico valores SO ₂ . Fuente: Elaboración propia	97
Tabla 6.17. Panel, valor máximo CO. Fuente: Elaboración propia	98
Tabla 6.18. Panel, valor mínimo CO . Fuente: Elaboración propia.....	98
Tabla 6.19. Panel, valor promedio CO. Fuente: Elaboración propia.....	99
Tabla 6.20. Panel, gráfico valores CO. Fuente: Elaboración propia.....	99
Tabla 6.21. Panel, valor máximo PM ₂₅ . Fuente: Elaboración propia.....	101
Tabla 6.22. Panel, valor mínimo PM ₂₅ . Fuente: Elaboración propia.....	101
Tabla 6.23. Panel, valor promedio PM ₂₅ . Fuente: Elaboración propia	102
Tabla 6.24. Panel, gráfico valores PM ₂₅ . Fuente: Elaboración propia	102
Tabla 6.25. Panel, valor máximo PM ₁₀ . Fuente: Elaboración propia	103
Tabla 6.26. Panel, valor mínimo PM ₁₀ . Fuente: Elaboración propia.....	104
Tabla 6.27. Panel, valor promedio PM ₁₀ . Fuente: Elaboración propia.....	104
Tabla 6.28. Panel, gráfico valores PM ₁₀ . Fuente: Elaboración propia	105
Tabla 6.29. Panel, valor máximo O ₃ . Fuente: Elaboración propia.....	106
Tabla 6.30. Panel, valor mínimo O ₃ . Fuente: Elaboración propia	107
Tabla 6.31. Panel, valor promedio. Fuente: Elaboración propia	107
Tabla 6.32. Panel, gráfico valores O ₃ . Fuente: Elaboración propia	108

Capítulo 1

1. Introducción

En este apartado se presenta el contexto general en el que se enmarca este proyecto y la motivación que ha llevado a su desarrollo. Además, se definen los objetivos principales, se delimita su alcance y se describe la estructura bajo la cual se organizará el contenido.

1.1 Contexto y motivación

En los últimos años, la preocupación por el cambio climático y efectos de la calidad del aire ha crecido notablemente, y con razón, ya que los índices de contaminación han ido en crecimiento en la mayoría de regiones mundiales. Esta situación favorece el desarrollo de tecnologías para la monitorización ambiental.

La presencia de contaminantes como las partículas PM_{2.5}, representan una amenaza directa para la salud mundial. Estos contaminantes, suspendidos en el aire, pueden penetrar profundamente en los pulmones debido a su diminuto tamaño, provocando efectos adversos incluso a bajas concentraciones.

En la figura 1.1 un mapa global con los niveles estimados de PM_{2.5} y PM₁₀, representados por colores según su concentración, más oscuro cuanto más concentración se encuentra. Esta imagen ejemplifica la gravedad y la distribución geográfica del problema.

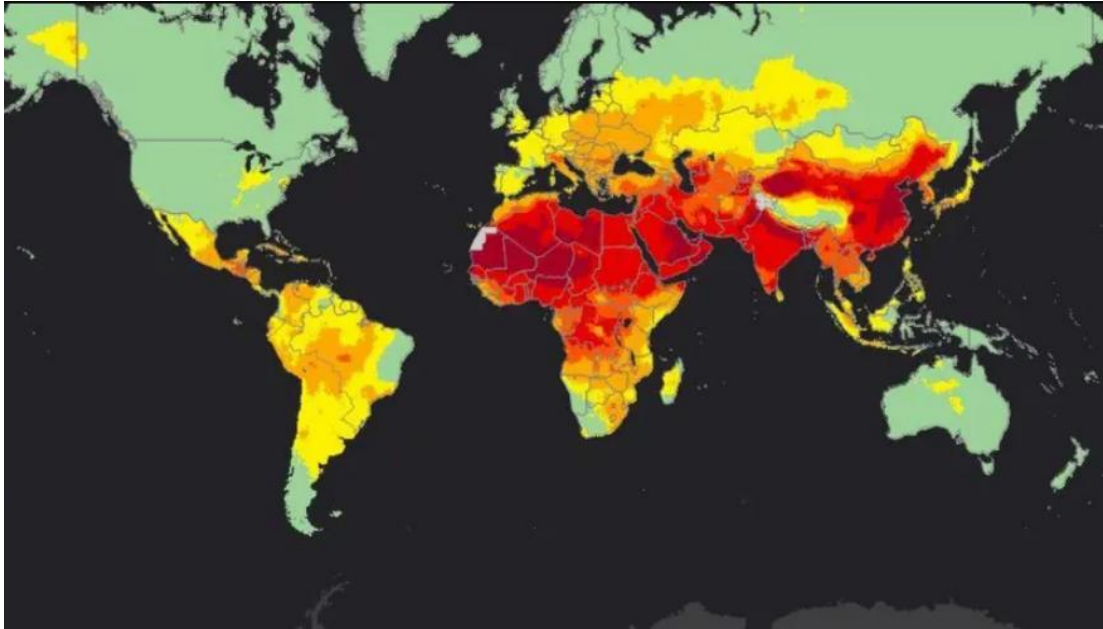


Figura 1.1. Mapa contaminación del aire a nivel global [1].

Contar con sistemas de monitorización ambiental u otro tipo de avances, puede ayudar a mejorar la gestión y anticiparse a situaciones alarmantes, generar conciencia social y hábitos sostenibles a largo plazo y facilitar el estudio del clima pudiendo acceder a un histórico con los datos necesarios. Este tipo de soluciones resulta muy importante, incluso esenciales para instituciones públicas, investigadores o ciudadanos de a pie.

La recogida y análisis de datos ambientales se ha visto reforzada en los últimos años gracias al auge del Internet de las Cosas (IoT) y al crecimiento de plataformas capaces de procesar grandes volúmenes de datos en tiempo real. Esta evolución ha permitido crear sistemas capaces de recopilar información desde diversas fuentes y transformarla en conocimiento útil para la toma de decisiones. A su vez, el avance de las tecnologías de código abierto ha facilitado la creación de sistemas distribuidos, escalables y fácilmente desplegables. Herramientas como Apache NiFi, InfluxDB, MinIO o Grafana permiten construir soluciones completas para la extracción, almacenamiento y visualización de grandes conjuntos de datos.

Este proyecto se crea con la motivación de aplicar estas tecnologías en un caso práctico, trabajando sobre los datos de calidad del aire y temperaturas en la comunidad de Castilla y León. Se busca no solo consolidar muchos conocimientos adquiridos durante la carrera, sino también

desarrollar una solución funcional, automatizada y modular que permita almacenar y visualizar la evolución de los datos ambientales, de forma sencilla mediante gráficos interactivos.

Además, se opta por la utilización de tecnología dedicada a la organización y separación de servicios, para la creación de una solución modular, eficiente, estandarizada y escalable. Se conseguirá de esta manera la capacidad de efectuar un despliegue de todas las funcionalidades del sistema de manera independiente, facilitando así su instalación, mantenimiento y portabilidad.

1.2 Objetivos

El trabajo por realizar consistirá en el diseño e implementación de un sistema de almacenamiento para la gestión y manejo de datos. Para ello el sistema incorporará diferentes elementos y estructuras necesarios para la ingesta, transformación, limpieza y almacenamiento de datos. El conjunto se integrará utilizando contenedores independientes para que los elementos anteriores trabajen en conjunto y con un alto nivel de escalabilidad. Los objetivos principales que se plantean en el TFG son los siguientes:

- Creación de entorno centralizado de almacenamiento tipo Datalake.
- Configuración e implementación de un sistema para la ingesta de datos desde diferentes orígenes.
- Despliegue de un servidor preparado para el almacenamiento de datos.
- Disposición de los elementos de la arquitectura en contenedores independientes, para obtener un sistema totalmente modular.
- Integración de una herramienta de visualización para la interpretación de la información guardada utilizando gráficos comprensibles.

1.3 Alcance del proyecto

El alcance de este trabajo será especificado en los puntos a continuación. Se ha tomado la decisión del desarrollo teniendo en cuenta unos márgenes debido a la amplitud y la extensión que podría llegar a tener un sistema de esta índole, siendo modular y fácilmente escalable:

- El proyecto se centrará exclusivamente en la obtención, almacenamiento y visualización de datos ambientales (calidad del aire y temperaturas) de la comunidad de Castilla y León.
- El sistema a desarrollar se limitará al uso en local gestionado y organizado en contenedores Docker, sin despliegue en la nube.
- Se utilizarán herramientas y servicios de código abierto: MinIO, Apache Nifi, InfluxDb, Grafana y Docker.
- No se aplicarán técnicas de aprendizaje automático ni predicciones de ningún tipo.
- Se incluirá el flujo completo de datos desde su extracción hasta su visualización.
- El sistema no contempla la utilización de dispositivos físicos, como sensores de ningún tipo.

1.4 Distribución del contenido

La estructura mediante la cual se organizará el documento tendrá diferentes secciones divididas a su vez de subsecciones, para ordenar el documento de la forma más clara, ordenada y accesible. Se seguirá la siguiente distribución:

Sección 1: Introducción. Se contextualiza el entorno y la problemática actual por contaminación. En este apartado se expondrán los objetivos y alcance del proyecto.

Sección 2: Marco teórico. Se recogen los fundamentos teóricos necesarios para la correcta comprensión del trabajo. Se introducen las tecnologías empleadas y los conceptos relacionados con la calidad del aire, proporcionando la base conceptual sobre la que se apoya el desarrollo del sistema.

Sección 3: Arquitectura del sistema. En esta sección se describe la arquitectura global del sistema, detallando su organización y la interacción entre servicios. Se presta especial atención al modelo basado en contenedores.

Sección 4: Análisis y diseño. Este apartado aborda el análisis previo de datos y su modelado, así como el diseño del flujo que siguen a lo largo del sistema. Se definen las estructuras necesarias y se establecen las decisiones de diseño que guían las fases posteriores de implementación.

Sección 5: ETL. Se describe el proceso completo de extracción, transformación y carga de los datos. Se detallan las distintas fases que permiten obtener la información desde su origen, procesarla adecuadamente y dejarla disponible para su posterior explotación.

Sección 6: Exploración. Se centra en la visualización y exploración de los datos almacenados. Se describe la planificación de las visualizaciones, su implementación y los mecanismos de interacción que permiten al usuario analizar la información de forma dinámica e intuitiva.

Sección 7: Conclusiones y trabajo futuro. Se presentan las conclusiones obtenidas del trabajo, evaluando el cumplimiento de los objetivos propuestos. También se plantean posibles trabajos futuros orientados a la mejora y ampliación del sistema desarrollado.

Capítulo 2

2. Marco teórico

2.1 Tecnologías empleadas

En la siguiente sección se describirán las principales tecnologías utilizadas en la elaboración del sistema. Cada una desempeñará una labor específica para el transcurso del flujo de datos y para la funcionalidad general del sistema. En los apartados siguientes se mencionan brevemente las propiedades, competencias y justificación de su uso en este proyecto. Las herramientas siguientes han sido seleccionadas tras un análisis exhaustivo de los objetivos a alcanzar, teniendo muy en cuenta la cooperación y la correcta integración de todas ellas para un óptimo funcionamiento en conjunto.

2.1.1 Docker

Docker [2] es una herramienta de código abierto desarrollada para la virtualización mediante la creación, gestión y ejecución de contenedores. Un contenedor es una unidad ligera o portátil que permite contener todo lo necesario para poder desplegar una aplicación, incluyendo código, librerías y dependencias de una manera aislada del sistema. Estos contenedores utilizan pocos recursos y de esta manera se conseguirá una rápida ejecución.

Docker se diferencia de las máquinas virtuales en que Docker solo virtualiza la capa de aplicación mientras que las VM (Virtual Machine) también lo hacen con la capa kernel. Docker utiliza el kernel del sistema operativo anfitrión. Gracias a ello Docker ocupa poco y es bastante veloz, en comparación con las VM que utilizan más recursos y son más pesadas y lentas. En el diagrama situado en la parte inferior se muestra de manera visual y sencilla las capas principales de un sistema y la comparación en cuanto a virtualización, de Docker con una máquina virtual tradicional. Podemos ver claramente cómo y por qué Docker tendrá una menor demanda de recursos incrementando la velocidad y la eficiencia.

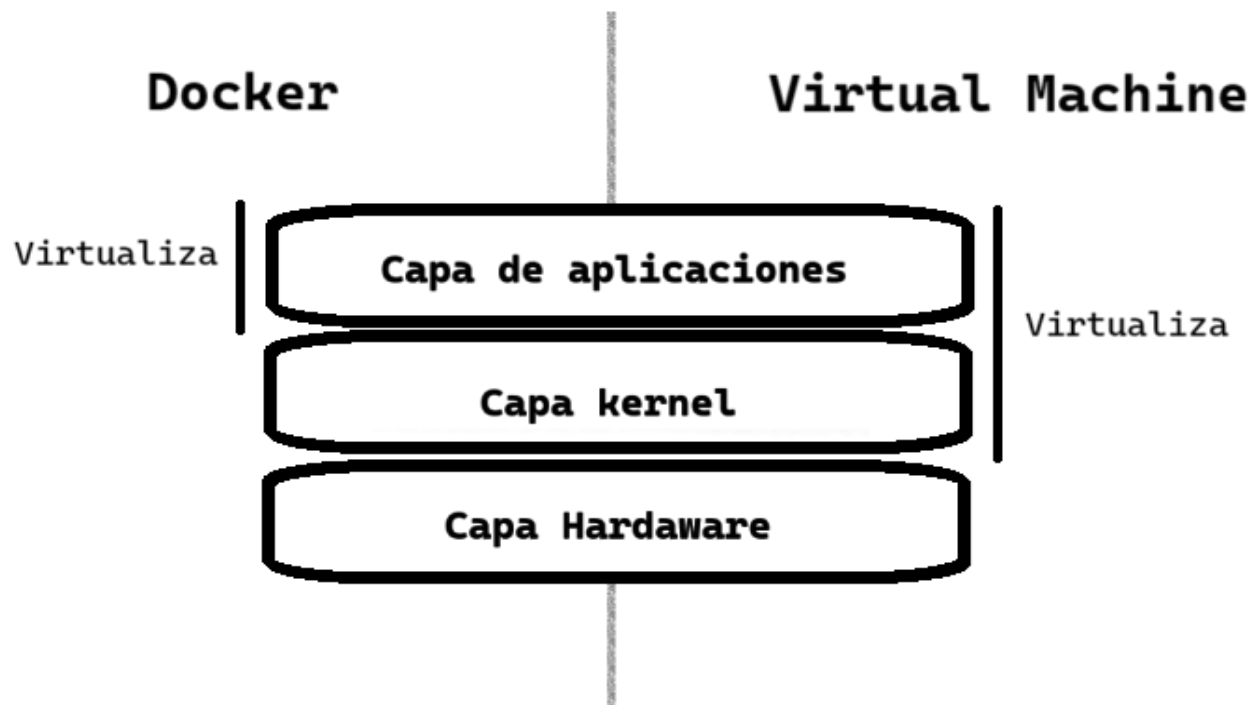


Figura 2.1. Comparativa de Docker y VM. Fuente: Elaboración propia

Para un arranque sencillo y una gestión conjunta de los servicios, se emplea Docker Compose [3], una herramienta que permite definir y ejecutar múltiples contenedores mediante un único archivo YAML. Con este archivo se permitirá desplegar todo el sistema de forma coordinada con un solo comando:

```
\Escritorio\DataLake_TFG\Project> docker-compose up -d
```

Figura 2.2. Comando en PowerShell para levantar el docker-compose. Fuente: Elaboración propia

Al ejecutar el comando dentro de la carpeta donde se encuentra el archivo .yaml de Docker compose, se consigue que todos los servicios que se encuentran declarados dentro del archivo arrancarán con las configuraciones determinadas que hayamos plasmado en el archivo de Docker compose.

En la siguiente imagen visualizamos mediante un esquema de la estructura que tiene el docker compose implementado en este sistema, pudiendo contener varios contenedores independientes entre sí, y pudiendo ejecutarlos al mismo tiempo. Esta estructura nos permite que, aun siendo estas unidades aisladas e independientes, puedan comunicarse entre sí, al pertenecer todas a este Docker Compose.

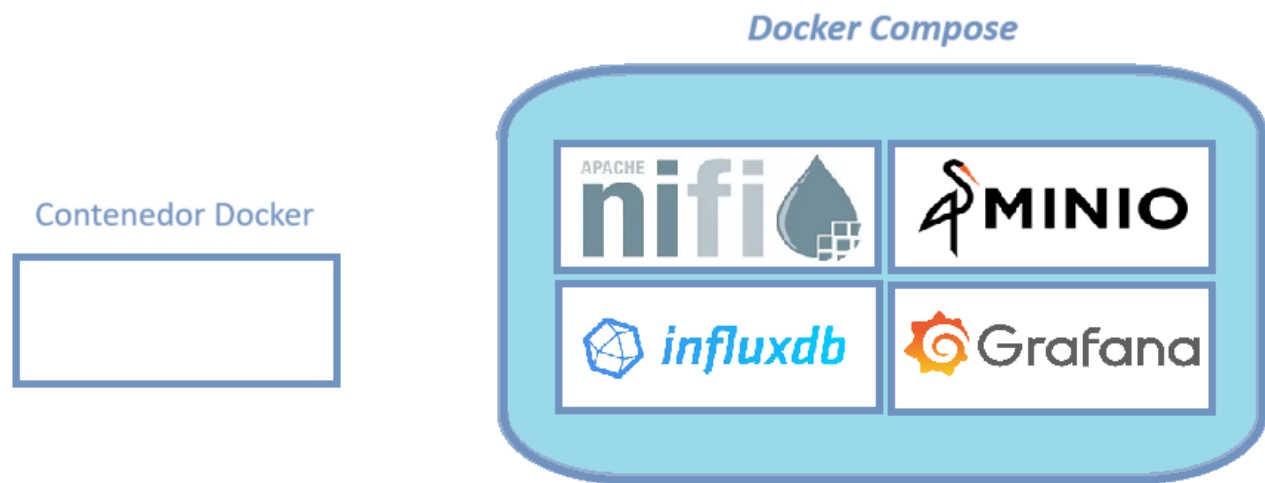


Figura 2.3. Esquema de estructura utilizada para Docker Compose. Fuente: Elaboración propia

2.1.2 Apache Nifi

Apache Nifi [4] [5] es otra de las herramientas que se utiliza en este trabajo, incluyéndose en un contenedor Docker para su despliegue. Se puede definir como software de integración de datos de código abierto, en el que se podrá automatizar el flujo de datos entre sistemas de una forma segura y escalable.

Los componentes más importantes y destacables de Nifi son los flujos de datos, los archivos de flujo y los procesadores [6]. Los flujos de datos se componen de varios pasos e instrucciones, creando una especie de trayectoria que seguirán los datos al moverse entre las diferentes partes de sistemas. Un archivo de flujo o Flow File representa un registro de datos que recorren el flujo, incluyen un contenido (los propios datos) y atributos (metadatos asociados). Los procesadores dentro de Nifi son los bloques de construcción básicos para construir un flujo. Son

componentes que se dedican a realizar funciones particulares dentro de la trayectoria de los datos. Existen procesadores para la ingestión, transformación, conexión, publicación, etc... de datos y se permitirá la implementación de procesadores personalizados. Se permite la unión de varios procesadores pudiéndose crear relaciones entre ellos de diferentes tipos.

Toda esta funcionalidad hace de Apache Nifi una herramienta óptima para la captura, filtrado, transformación, agrupación y envío de datos, dotando de una gran escalabilidad y eficiencia al sistema. Mediante la creación de flujos de datos, se capta información de la temperatura y calidad del aire en Castilla y León, y mediante una serie de procesadores se adapta y estructura, para ser almacenada en el servidor. Tras ello los datos serán normalizados y enviados a InfluxDB, para su posterior presentación.

En términos generales se puede afirmar que Apache Nifi es el elemento integrador y puente de enlace entre los distintos servicios dockerizados, permitiendo su coordinación, representando una pieza clave para la orquestación.

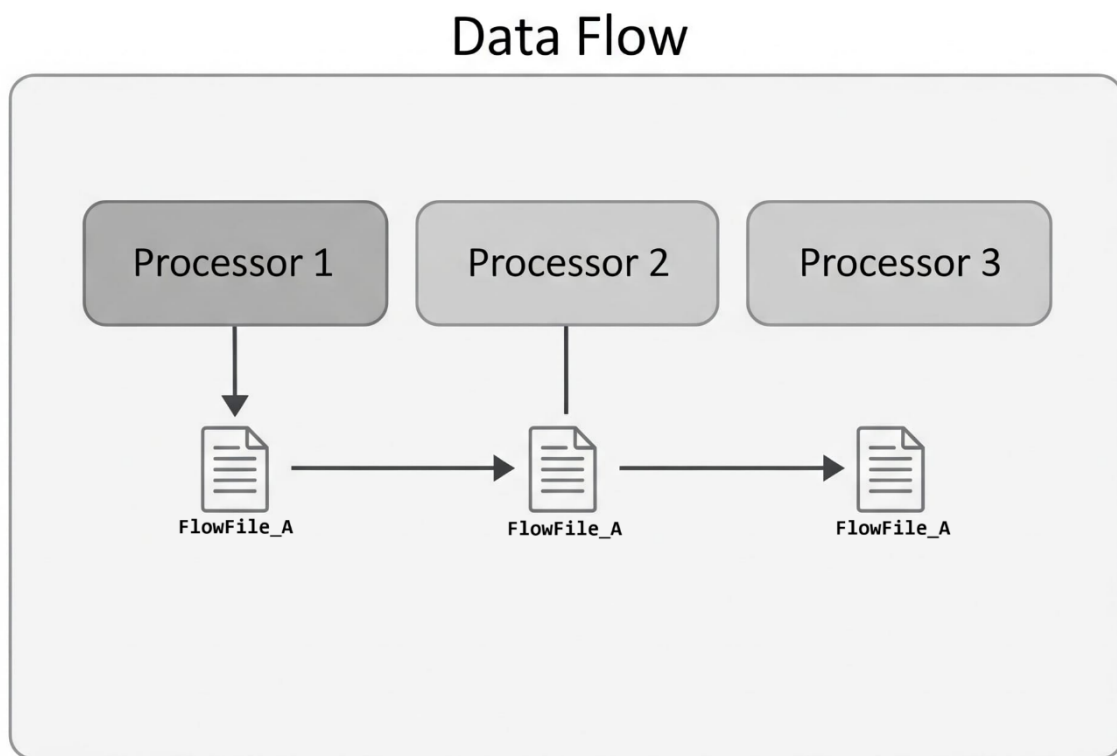


Figura 2.4. Funcionamiento flujo de datos Apache Nifi. Fuente: Elaboración propia

2.1.3 MinIO

MinIO es una plataforma para el almacenamiento y gestión de objetos, como archivos, documentos, imágenes [7]. Es un software de código abierto con una gran capacidad para escalar, compatible con Amazon S3. MinIO está desarrollado bajo la premisa de obtener un gran rendimiento para grandes volúmenes de datos en un entorno seguro y protegido.

El almacenamiento puede ser tratado de varias formas. Una de las maneras es guardar archivos organizados en carpetas, muy óptima para sistemas tradicionales. También se puede almacenar en bloques, donde los datos se guardan en fracciones de un tamaño concreto y fijo, ideal para algunas bases de datos. MinIO, sin embargo, utiliza objetos como modo de persistencia, donde cada dato se guarda con su propio contenido, metadatos asociados y un identificador, siendo éste único y siendo perfecto para la persistencia de grandes cantidades de datos sin estructura definida.

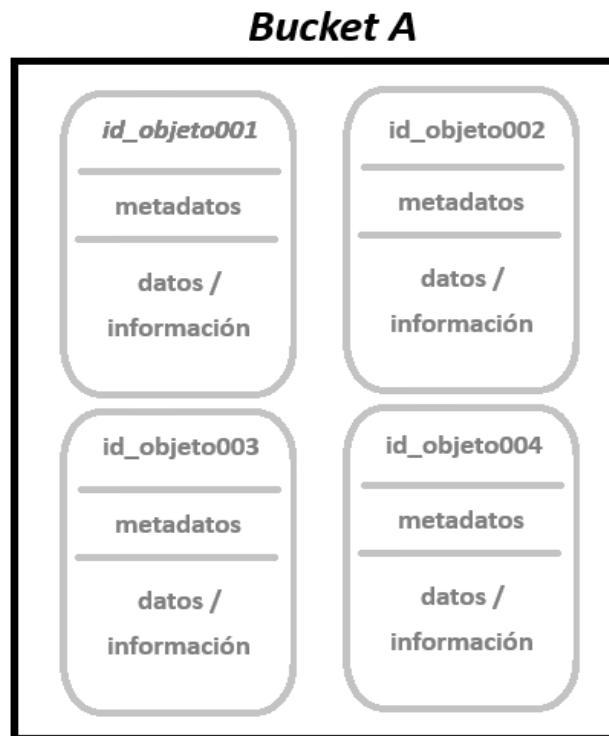


Figura 2.5. Organización de los objetos en MinIO. Fuente: Elaboración propia.

En MinIO los datos se estructuran a través de los buckets [8]. Cada bucket actúa como un contenedor lógico, donde se encuentran los objetos. Con su utilización se simplifica el acceso a la información, la recuperación y el almacenamiento seguro.

Se ha utilizado MinIO como sistema de almacenamiento para este proyecto debido a las ventajas que posee, entre las cuales están: es un software de código abierto y ligero, alto rendimiento, integración sencilla con tecnología Docker, persistencia segura, flexibilidad y gran escalabilidad.

Este software, actúa dentro del sistema modularizado como DataLake centralizado del proyecto. Se reciben los datos desde Apache Nifi, guardando la información y creando así un histórico de de datos ambientales y meteorológicos. Garantizamos también persistencia y fácil accesibilidad a los datos. En MinIO quedarán también copias de los datos más en crudo (antes de su total transformación), por si fuera necesario aplicar futuros procesos que requieran la información en formato original.,

Se despliega un servidor MinIO desde un docker-compose, junto al resto de servicios explicados con anterioridad [9]. Esto dotará al sistema de una gran escalabilidad. Desde un comando podremos levantar todos los servicios que se especifiquen dentro del archivo docker compose, éste incluido.

Teniendo en cuenta la explicación anterior, se obtiene la justificación de por qué MinIO es una herramienta perfecta para el funcionamiento de este sistema.

2.1.4 InfluxDB

InfluxDB [10] es una base de datos orientada a series temporales cuyo enfoque está orientado a el almacenamiento y consulta de datos que varían en el tiempo. Esta herramienta es óptima para datos con marca temporal, soportando grandes inserciones de datos. Tiene su propio lenguaje de consulta específico (FLux/InfluxQL) [11].

Para su funcionamiento básico debemos tener en cuenta los conceptos clave:

1. El measurement actúa como el nombre de la métrica o del conjunto de datos, en nuestro caso, por ejemplo, temperatura o aire.
2. Los tags son pares de clave - valor que se usan como etiquetas de indexación para la información, que permiten búsquedas rápidas y eficientes (por ejemplo, provincia = Segovia).
3. Los fields representan los valores medidos que suelen ser tanto en forma de texto como numéricos (por ejemplo, NO₂ = 25.3),

Cada registro tendrá también una marca de tiempo o timestamp, que nos indicará el instante exacto en el que se realizó la medición, lo que nos dejará analizar los datos a lo largo del tiempo. Un registro de InfluxDB responde a la estructura de line protocol [12], en la que encontramos los apartados descritos anteriormente. En la siguiente imagen vemos la estructura de un registro de la base de datos Influx de este sistema, para aterrizar los conceptos tratados, en formato line protocol, que consiste en una línea de texto donde encontramos todos los elementos explicados. Sin rodear, en el ejemplo, encontramos los campos o fields, descritos anteriormente, donde encontramos el contaminante y su concentración.

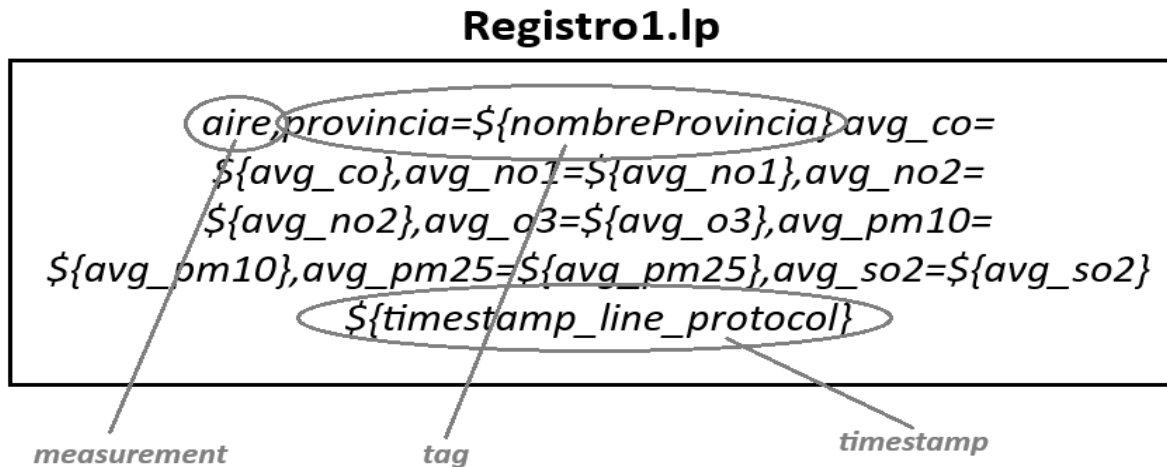


Figura 2.6. Formato Line Protocol. Fuente: Elaboración propia.

Los datos se organizan en series temporales, estableciendo un conjunto todos lo que posean el mismo measurement, por ejemplo, en este caso todos los que sean measurement igual a aire. Esta serie contendrá un número fijo de tags, múltiples campos y marcas de tiempo. Estas series se organizan dentro de buckets, que permiten separar o delimitar diferentes conjuntos de información, haciendo más fácil la gestión de registros.

En cuanto a las propiedades ventajosas que InfluxDB nos aporta en este proyecto, tenemos que esta herramienta está especializada en series temporales, siendo bastante eficiente y de código abierto. Tiene una gran escalabilidad horizontal y su ecosistema es ampliable mediante servicios como Telegraf o Kapacitor, por si en un futuro se desea incluir o ampliar alguna de sus competencias. La compatibilidad multiplataforma y su sencillo despliegue mediante contenedores Docker, será de gran ayuda, al igual que su documentación y su amplia comunidad para soporte. Su integración nativa con Grafana será un factor crucial, ya que será otra de las herramientas que se utilizará en el sistema y se explicará más adelante.

InfluxDB desempeña un importante papel dentro del sistema, almacenando la información en formato de series temporales. Los datos una vez transformados y estructurados con Apache Nifi serán enviados mediante HTTP a Influx, donde quedarán organizados en measurements, con sus correspondientes tags, fields y timestamps. Su integración directa con Grafana hará de InfluxDB,

la fuente de datos principal para la visualización, aportando una fácil gestión y datos actualizados para su posterior análisis.

2.1.5 Grafana

Grafana es una plataforma open source para la visualización, presentación y monitorización de datos, orientada a la creación de paneles interactivos y personalizables. Tiene un amplio soporte de fuentes de datos, entre las que se encuentra InfluxDB [13].

Esta herramienta permite desarrollar e instaurar dashboards dinámicos, posibilitando la representación de gráficas, tablas, mapas, etc. En un mismo panel se tiene la opción de utilizar varias fuentes de datos diferentes, dotando a esta capa del sistema de mucha escalabilidad y múltiples opciones de personalización. Su interfaz es bastante intuitiva y es accesible vía web.

Grafana, no almacena los datos por sí misma, si no que utiliza fuentes externas. La conexión se realiza configurando un datasource en la interfaz, donde se indican credenciales, direcciones y otros parámetros de acceso.

Para obtener la información de la fuente, Grafana utiliza el lenguaje nativo o propio de base de datos utilizada, en este caso InfluxQL, que permite seleccionar y filtrar por rangos de tiempo, agrupar por intervalos y aplicar otra serie de funciones [14].

A la hora de representar visualmente la información se podrá elegir la herramienta que más se adapte a la circunstancia. En la siguiente figura podremos observar algunas de las más importantes, mediante las cuales se representarán los datos anteriormente obtenidos. Estas representaciones serán configurables, seleccionando escalas de tiempo, colores, etiquetas, umbrales y leyendas. El usuario final a su vez podrá interactuar con los paneles, comparar periodos o aplicar filtros, entre otras funcionalidades.

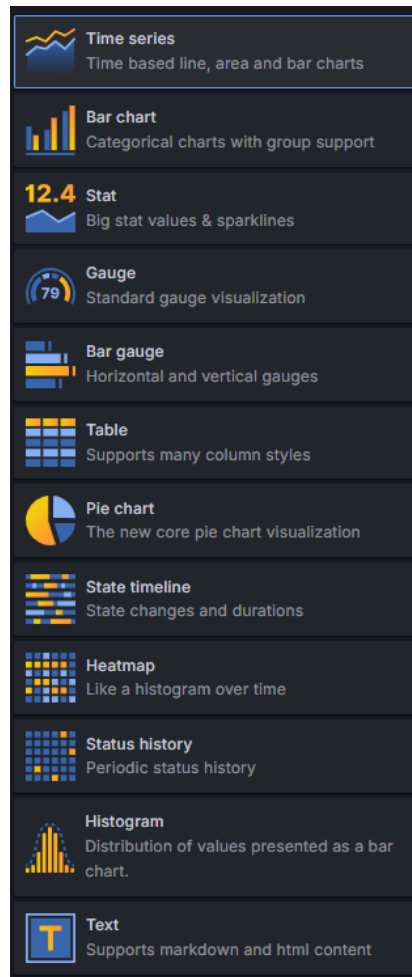


Figura 2.7. Tipos de panel de visualización Grafana. Fuente: Interfaz de control Grafana.

Grafana constituye la capa final del sistema donde todos los datos confluyen y convergen, pudiendo mostrar una gran cantidad de información de una forma estética, ordenada y comprensible. Gracias a esta plataforma podemos detectar patrones, anomalías o tendencias, que de otra manera sería bastante dificultoso de revelar. Al tener un alto grado de interactividad y accesibilidad vía web, se logra que esta herramienta sea perfecta para este sistema, utilizando series temporales y datos cronológicos de una manera óptima y sencilla. La centralización de toda la información de esta manera, hace que Grafana se convierta en el “tablero de control” del proyecto pudiendo integrar todo lo recogido mediante el complejo flujo de datos y visualizarlo de manera creativa y adecuada para cualquier usuario.

2.2 Control de la calidad del aire

El control de la calidad del aire resulta fundamental, ya que tiene un impacto muy relevante en la salud de la población global y el equilibrio del medio ambiente. En el caso de que la capa de gases que rodea el planeta, contenga altos niveles de contaminantes puede provocar múltiples efectos irreversibles en muchos ámbitos. Por ello existen diversos estándares y regulaciones y normativas para intentar solventar o minimizar el gran problema de contaminación del aire.

Para un correcto análisis y control se deberán abordar los factores clave que provocan la emisión de elementos contaminantes, ya sean industriales, tráfico, meteorología.... El objetivo de la monitorización no es únicamente medir, sino detectar problemas, análisis de tendencias y prevención de riesgos.

En este apartado se describirán los principales contaminantes, se presentarán consideraciones relevantes y el modo cómo se toman los datos de la condición del aire en las estaciones oficiales.

2.2.1 Elementos contaminantes

La calidad del aire se analiza y determina basándose en la concentración de diferentes sustancias contaminantes presentes en la atmósfera, que pueden tener múltiples efectos adversos. En este proyecto se trabajará con los principales compuestos contaminantes, que son objeto de las principales estaciones de medición oficiales. Entre estos se encuentran gases, partículas sólidas y elementos químicos derivados de procesos de diferente índole.

A continuación, se clasificará y se dará una explicación sencilla tratando perjuicios, naturaleza y proveniencia de cada contaminante monitorizado.

- ***Monóxido de carbono (CO):***

Gas incoloro e inodoro generado principalmente por combustiones incompletas cuyo origen pueden ser los vehículos, las calefacciones o estufas de gas, chimeneas,

procesos industriales... Este compuesto, puede afectar a la oxigenación sanguínea, uniéndose a la hemoglobina. Realmente la hemoglobina de la sangre tiene mucha más afinidad con el monóxido de carbono que con el propio oxígeno (O_2), por lo cual, existiendo altas concentraciones de este gas, se bloquea el transporte de oxígeno a los tejidos y puede ser muy perjudicial [15].

- ***Monóxido de nitrógeno (NO):***

Gas incoloro e inodoro cuya naturaleza es poco estable y reactiva, ya que se oxidará fácilmente en presencia de oxígeno, formando otro de los gases que analizaremos posteriormente NO_2 . El NO se produce en procesos donde existe combustión a altas temperaturas, en tormentas eléctricas y rayos, y en procesos biológicos de ciertos microorganismos, o incluso dentro de nuestro intestino. No es demasiado tóxico de por sí mismo, pero sí lo es cuando se convierte en NO_2 , por ello es de vital importancia la monitorización de sus niveles [16].

- ***Dióxido de Nitrógeno (NO_2):***

El NO_2 es un gas irritante de color marrón-rojizo proveniente de la oxidación de el monóxido de nitrógeno, cuyo olor es picante y penetrante, pudiendo ser percibido incluso en concentraciones muy bajas. Este compuesto se disuelve en el agua, pudiendo darse así el fenómeno de la lluvia ácida. Este gas genera irritación en la garganta, nariz, ojos y vías respiratorias, además, una exposición prolongada puede empeorar el asma, bronquitis o enfermedades pulmonares, pudiendo hasta causar daños a estos tejidos. Este compuesto participa en la formación de ozono troposférico (O_3) [16].

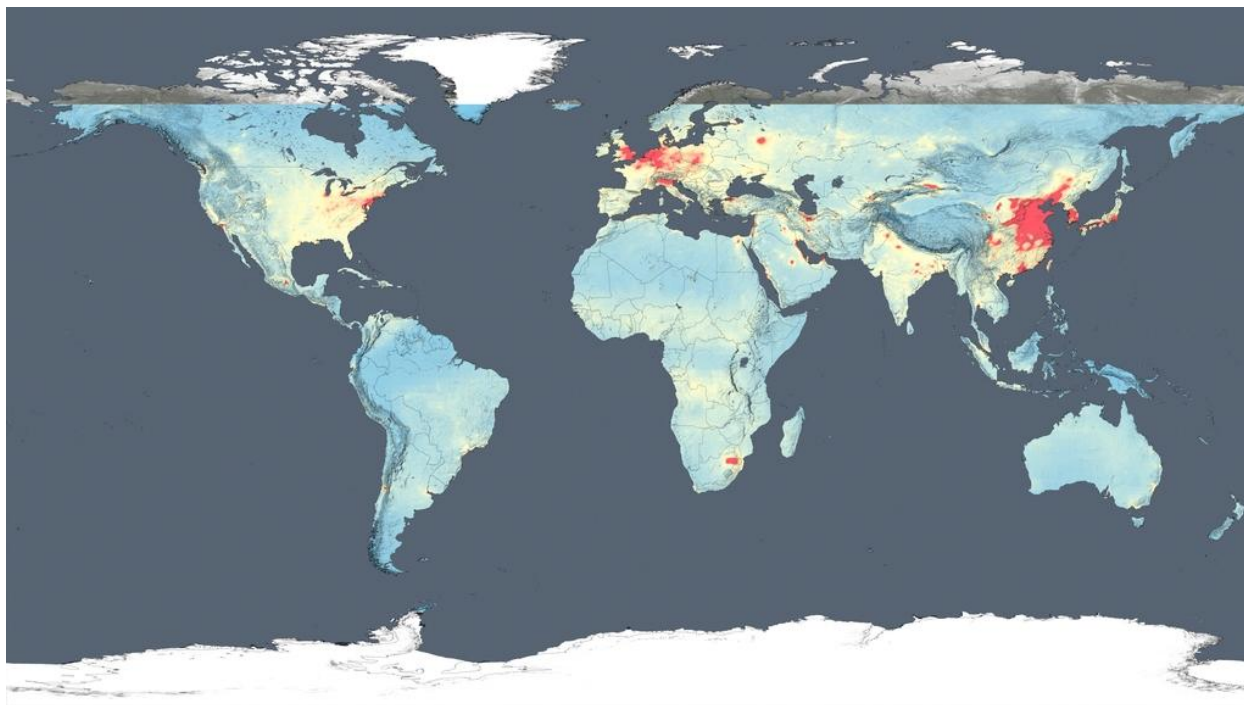


Figura 2.8. Mapa global de concentración NO2 [17].

- ***Ozono troposférico (O3):***

El Ozono es un gas que se encuentra de manera natural en la atmósfera. Posee un color azul pálido con un olor penetrante algo dulce. A temperatura ambiente es muy reactivo, aunque es poco soluble en agua, es muy oxidante. Este elemento no se emite de una forma directa, sino que se forma a raíz de reacciones fotoquímicas con otros elementos como el propio NO o NO2 expuestos anteriormente. Este gas es beneficioso a nivel estratosférico, pero en la troposfera, la capa más cercana a la superficie terrestre, es considerado un contaminante. Causa irritación en los ojos, nariz, garganta y vías respiratorias. Es muy peligroso para niños, ancianos y personas con problemas respiratorios. También afecta negativamente a la vegetación y a los cultivos [18].

- ***Dióxido de azufre (SO₂):***

Este elemento es también un gas a temperatura ambiente, incoloro, pero con un fuerte olor. Es soluble en agua y se produce con la combustión de combustibles fósiles que contienen azufre, como son el carbón, el petróleo y otros derivados. El dióxido de azufre causa irritación en las zonas más delicadas de la cara, pudiendo causar problemas respiratorios y daños pulmonares. Al ser soluble en agua, contribuye a la lluvia ácida, pudiendo dañar la vegetación, los suelos y las construcciones [19].

- ***Partículas en suspensión menores de 10 µm (PM10):***

Mezcla de partículas tanto sólidas como líquidas cuyo diámetro resulta inferior 10 a micrómetros (µm). Sus orígenes pueden ser naturales como pueden ser, el polvo del suelo, polen, esporas, arena o cenizas. Su procedencia también pueden ser emisiones de vehículos, industria, combustión o la propia construcción. Al ser inhaladas este material llega principalmente a la parte superior de las vías respiratorias. Puede causar la irritación de las zonas más sensibles y se pueden depositar en suelos y en las fuentes de agua, modificando y pudiendo afectar a diversos ecosistemas [20].

- ***Partículas en suspensión menores de 2.5 µm (PM2.5):***

Estas son partículas con diámetro menor a 2.5 µm y proceden de la combustión de materiales fósiles, procesos industriales y otro tipo de formaciones secundarias y reacciones químicas. A diferencia de las anteriores este tipo de material particulado penetra profundamente en las vías respiratorias hasta los alvéolos incluso pudiendo entrar al torrente sanguíneo. Se asocian, a menudo, con enfermedades cardiovasculares, infartos y enfermedades respiratorias graves y muy graves. Una exposición prolongada aumenta la mortalidad prematura, al igual que las PM10. En la figura 2.9 se observan las zonas que más exposición tienen a este tipo de partículas, siendo la franja del ecuador en la que más concentración existe, variando a lo largo del tiempo debido a diversos factores que se explicarán posteriormente [20].

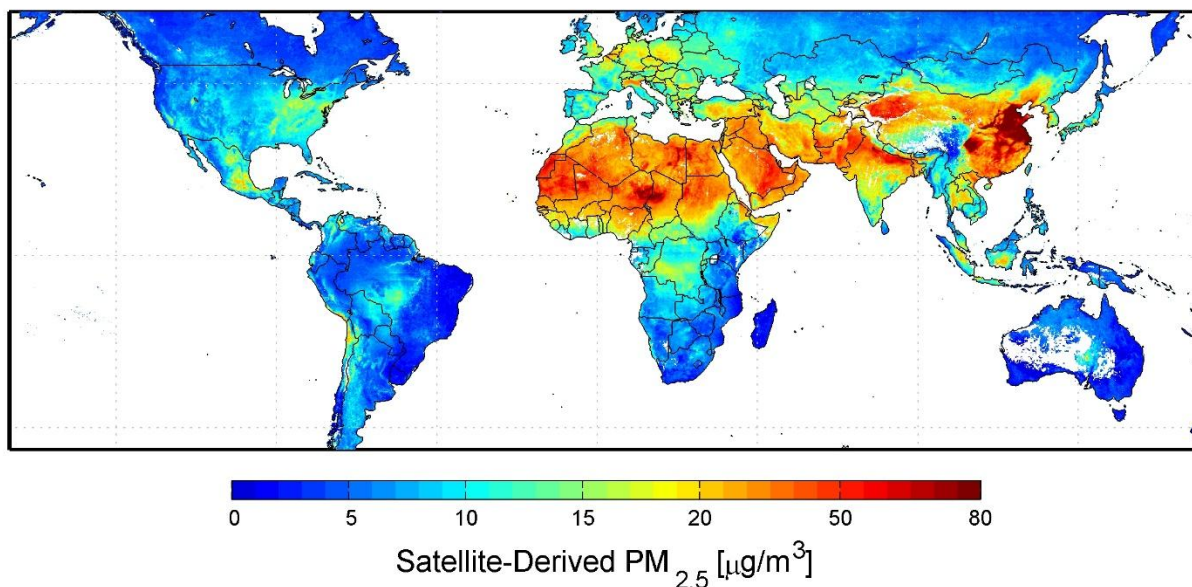


Figura 2.9. Mapa global de material particulado 2.5 [21]

Tras la breve explicación de cada uno de los elementos contaminantes y teniendo en cuenta que no son los únicos que pueden causar estragos, se puede afirmar que la monitorización de estos 7 contaminantes es muy importante y esencial, tanto para la prevención de daños, como para la protección de la salud y el medio ambiente.

2.2.2 Influencia de las condiciones meteorológicas

En este proyecto, la relación entre meteorología y contaminación posee una gran relevancia, ya que las condiciones climatológicas desempeñan un papel en la concentración, dispersión y transformación de los principales contaminantes. Factores como la temperatura, las precipitaciones, la humedad, la velocidad y dirección del viento o la radiación solar influyen directamente en la calidad del aire y en cómo estos contaminantes se comportan en la atmósfera. El análisis y comprensión de estas interacciones resultará muy útil a la hora de monitorear e interpretar los datos registrados sobre la calidad del aire.

El viento y las precipitaciones por ejemplo desempeñan un importante papel en cuanto a la dispersión y transporte de contaminantes y la lluvia como agente de limpieza atmosférica, arrastrando partículas y reduciendo temporalmente sus concentraciones. La humedad también puede contribuir a que pequeños materiales particulados, como el PM_{2.5} permanezcan en suspensión. La temperatura, por ejemplo, afecta tanto a procesos de creación como de dispersión. En días más cálidos la radiación solar favorece diversas reacciones fotoquímicas que originan contaminantes secundarios, como el ozono troposférico, como explicamos en el apartado anterior. En cambio, en días más fríos o con radiación menor, las masas de aire pueden quedar atrapadas bajo otras capas más cálidas (inversión térmica), lo que produce que la acumulación de pequeñas partículas y gases como el NO₂ o SO₂, aumente cerca de la superficie terrestre [22].

Con esta serie de ejemplos podemos afirmar que las concentraciones de contaminantes en la atmósfera no únicamente dependen de las emisiones, sino que también dependen muy significativamente de las condiciones meteorológicas.

En este trabajo a parte de la extracción y estudio de los datos relacionados con la calidad del aire, trabajaremos con uno de los factores meteorológicos más influyentes, la temperatura. De esta manera se podrán observar las correlaciones existentes entre las variaciones de temperatura y las concentraciones de los contaminantes, lo que aportará un valor añadido al análisis y visualización de la información extraída. Con la incorporación de estas variables térmicas se podrán analizar patrones ambientales a lo largo de un periodo de tiempo determinado. De esta manera este proyecto podrá servir de base para efectuar estudios predictivos y la toma de decisiones en cuanto a gestión ambiental.

2.2.3 Captura de datos

Los datos utilizados en este trabajo se capturan en redes de estaciones distribuidas por el territorio español, midiendo los parámetros contaminantes y meteorológicos. Estas estaciones pertenecen a las redes autorizadas autonómicas o estatales, por lo que los datos son oficiales y de primera mano. En la ilustración a continuación encontramos situadas en el mapa las estaciones de control, repartidas por el suelo nacional. En este proyecto se tendrán en cuenta únicamente los datos oficiales extraídos en las estaciones de Castilla y León.

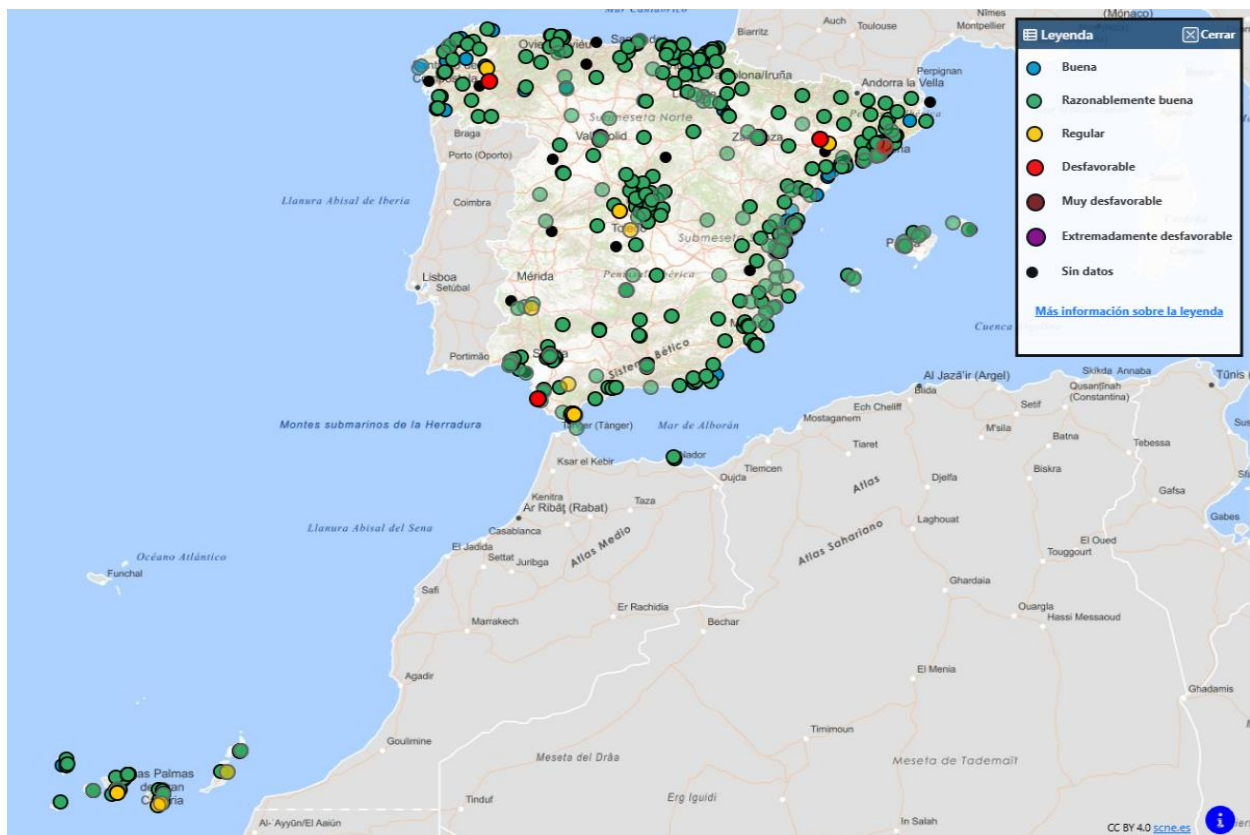


Figura 2.10. Distribución de las estaciones de control de calidad del aire en España [23].

Las mediciones que las estaciones destinadas a la calidad del aire obtienen, comprenden tanto los contaminantes gaseosos (NO, NO₂, SO₂, O₃, CO) como los materiales particulados (PM₁₀ y PM_{2.5}).

Capítulo 3

3. Arquitectura del sistema

3.1 Introducción a la arquitectura

El objetivo que se quiere conseguir con este sistema es el de recopilar, almacenar, filtrar, transformar y visualizar información ambiental y meteorológica de manera automatizada. Para ello se ha optado por una arquitectura modular y distribuida, para que ante todo sea un sistema escalable.

Se han utilizado los siguientes componentes clave para un completo y correcto funcionamiento:

- Apache Nifi: encargado del flujo de datos comprendiendo la ingesta y transformación. Desde Nifi se envían las solicitudes http tanto para los procesos de ingesta como para el envío de datos hacia minIO e Influx.
- MinIO: le corresponde el almacenamiento de datos en general, tanto en crudo, como ya transformados y preparados.
- InfluxDB: este componente constituye la base de datos perfecta para el tratamiento de series temporales.
- Grafana: es la herramienta utilizada para la visualización final.

A continuación, se representa un diagrama básico (*Figura 3.1*), de los componentes clave del sistema y sus fases en el flujo de datos, teniendo una primera visión en cuanto a la secuencia operativa del sistema de datos. Cada uno de los elementos principales mencionados anteriormente tendrá sus tareas definidas que trabajando en conjunto harán que los datos fluyan desde un origen en este caso web hasta la herramienta de visualización final.

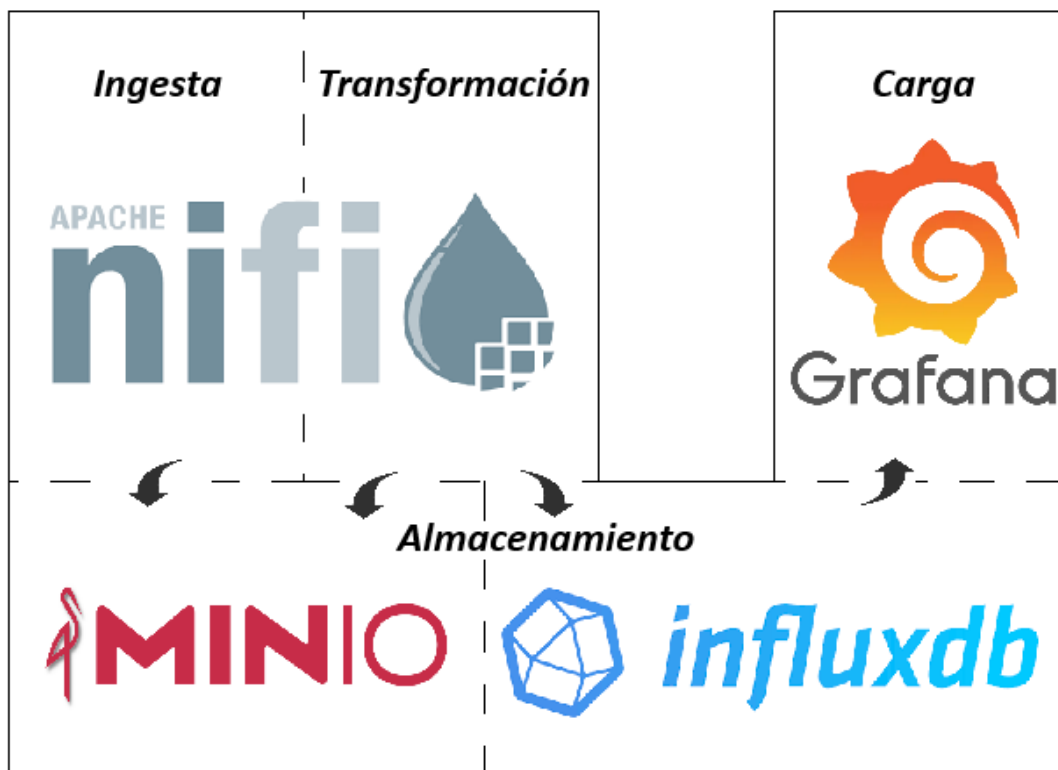


Figura 3.1. Diagrama del flujo de información entre los servicios. Fuente: Elaboración propia

3.1.1 Arquitectura dockerizada

Esta arquitectura está dockerizada, es decir implementa contenedores Docker para orquestar un correcto y sencillo despliegue. Docker es una herramienta de virtualización que se utilizará para el inicio y arranque de todos los servicios mencionados anteriormente. En el apartado 2.1.1 queda explicado su funcionamiento y cómo mediante un único comando, se ejecuta un docker compose, un archivo docker, que levanta y despliega todos los servicios utilizados de una forma eficiente. Esta potente herramienta ofrece portabilidad, aislamiento, facilidad de despliegue y un alto nivel de versatilidad. Docker se apoya en el sistema operativo para ejecutar los contenedores de una manera aislada con un uso de los recursos muy eficiente. Cada contenedor incluye las dependencias necesarias para su ejecución, compartiendo el kernel del sistema operativo del host entre todos ellos.

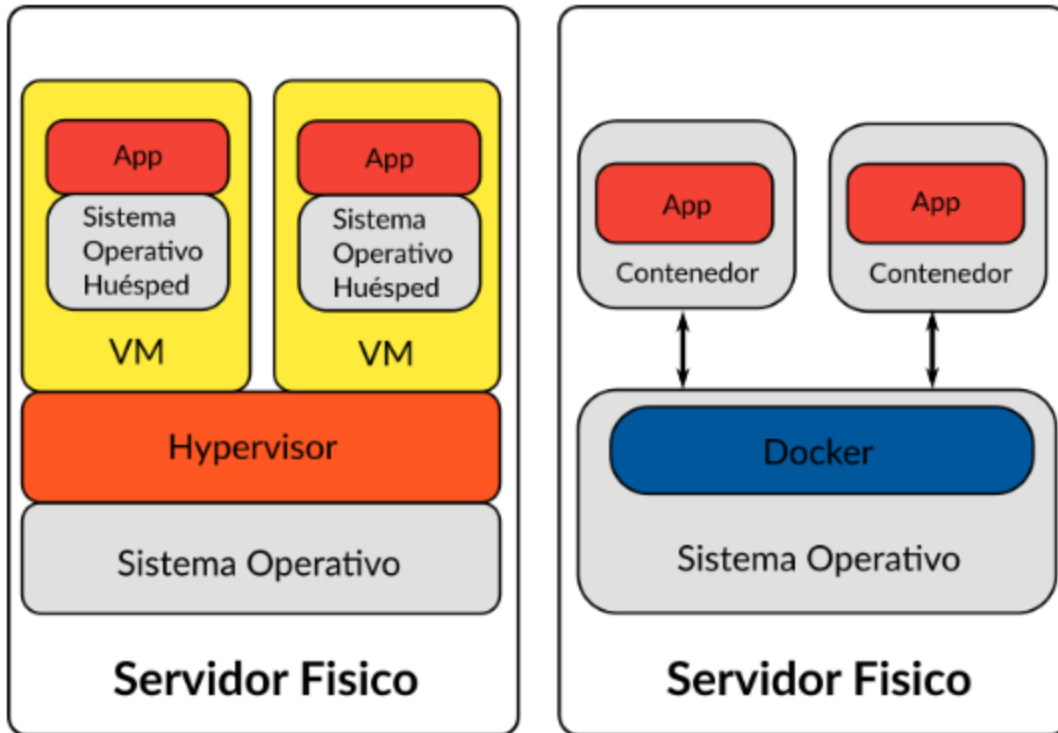


Figura 3.2. Arquitectura Docker [30].

3.2 Comunicación entre servicios

Es de vital importancia que todas las piezas del sistema se encuentren conectadas entre sí con el fin de garantizar el flujo continuo de datos desde el origen hasta el entorno de presentación. Todos los servicios, Apache Nifi, MinIO, InfluxDB y Grafana se encuentran contenedorizados e interconectados entre sí dentro de un entorno común orquestado por el Docker Compose.

3.2.1 Red interna de Docker

La configuración de Docker proporciona una red interna entre los componentes internos del Docker Compose, la cual, permite que todos los contenedores se comuniquen entre sí de forma directa y segura, utilizando el nombre del propio servicio a modo de identificador de red, sin la necesidad de trabajar con direcciones IP concretas.

Sobre la red descrita anteriormente se establecen las comunicaciones funcionales del sistema que permitirán que todos los servicios se encuentren conectados, para poder así hacer posible que el flujo de datos sea posible. Esta red virtual es creada automáticamente y será funcional para todos los servicios definidos en el archivo `docker-compose.yml`. A continuación, veremos una captura de pantalla del archivo `docker-compose.yml` donde se especifican las configuraciones necesarias para el correcto despliegue y funcionamiento del conjunto [24].

En el archivo, dentro de cada servicio, se define el nombre del servicio (que servirá de ID en la red interna), la imagen que Docker utilizará para correr el servicio, el nombre del contenedor que incluye el servicio (cada servicio su propio contenedor), el puerto por el que podremos acceder al servicio, los volúmenes necesarios para dotar de una funcionalidad completa al sistema y algunos otros ajustes adicionales requeridos. También se definirá el nombre de usuario y la clave que se utilizará para poder acceder a cada componente y a sus interfaces correspondientes.

```

1
2 services:
3   minio:
4     image: quay.io/minio/minio
5     container_name: ctMinio
6     ports:
7       - "9000:9000"
8       - "9001:9001"
9     command: server /data --console-address ":9001"
10    volumes:
11      - ./minio/data:/data # Volumen de persistencia de datos
12
13   nifi:
14     image: apache/nifi
15     container_name: ctNifi
16     ports:
17       - "8443:8443"
18     environment:
19       SINGLE_USER_CREDENTIALS_USERNAME: [REDACTED] # Usuario Nifi
20       SINGLE_USER_CREDENTIALS_PASSWORD: [REDACTED] # Contraseña Nifi
21     volumes: # A continuación los volúmenes de persistencia para datos y configuraciones de Nifi
22       - ./nifi/conf:/opt/nifi/nifi-current/conf
23       - ./nifi/database_repository:/opt/nifi/nifi-current/database_repository
24       - ./nifi/flowfile_repository:/opt/nifi/nifi-current/flowfile_repository
25       - ./nifi/content_repository:/opt/nifi/nifi-current/content_repository
26       - ./nifi/provenance_repository:/opt/nifi/nifi-current/provenance_repository
27       - ./nifi/input:/opt/nifi/nifi-current/input
28
29   influxdb:
30     image: influxdb:2.7
31     container_name: ctInflux
32     ports:
33       - "8086:8086"
34     environment:
35       - INFLUXDB_INIT_MODE=setup
36       - INFLUXDB_INIT_USERNAME=[REDACTED] # Usuario Influx
37       - INFLUXDB_INIT_PASSWORD=[REDACTED] # Contraseña Influx
38       - INFLUXDB_INIT_ORG=my-org
39       - INFLUXDB_INIT_BUCKET=my-bucket
40       - INFLUXDB_INIT_RETENTION=1h
41       - TZ=Europe/Madrid
42     volumes:
43       - influxdb_data:/var/lib/influxdb2 # Volumen de persistencia para los datos y configuraciones InfluxDB
44     restart: always
45
46   grafana:
47     image: grafana/grafana:latest
48     container_name: ctGrafana
49     ports:
50       - "3000:3000"
51     volumes:
52       - grafana-storage:/var/lib/grafana # Volumen de persistencia de Grafana
53       - /etc/localtime:/etc/localtime:ro # Se vincula la fecha y hora del host al contenedor
54     environment:
55       - TZ=Europe/Madrid
56       - GF_SECURITY_ADMIN_USER=[REDACTED] # Usuario Grafana
57       - GF_SECURITY_ADMIN_PASSWORD=[REDACTED] # Contraseña Grafana
58     depends_on:
59       - influxdb
60     restart: always
61
62 volumes:
63   grafana-storage:
64   influxdb_data:

```

Figura 3.3. Archivo docker-compose.yml. Fuente: Elaboración propia

3.2.2 Flujos de comunicación entre servicios

La red interna permite que todas las herramientas intercambien información mediante protocolos HTTP y las APIs concretas, según cada proceso. Los principales flujos de comunicación que se utilizan en este proyecto son:

- **Fuente web - Apache Nifi:** Mediante peticiones HTTP GET a las APIs oficiales de datos abiertos de Castilla y León Nifi captura los datos meteorológicos y de calidad del aire de forma automática.
- **Apache Nifi - MinIO:** Apache Nifi envía los datos en crudo o ya transformados y mapeados al Data Lake mediante la API S3, que MinIO respalda, en forma de archivos de tipo JSON. Estos procesos garantizan que la información esté siempre almacenada y respaldada, pudiendo ser utilizada posteriormente.
- **Apache Nifi - InfluxDB:** Por medio de solicitudes HTTP POST se irán enviando los datos desde Nifi a nuestra base de datos de series temporales, empleando el formato Line Protocol, para una fácil interpretación en el destino. Se efectuará una solicitud por provincia y tipo de información. En el apartado 2.1.4 se expone un ejemplo de este formato, que InfluxDB leerá.
- **Grafana - InfluxDB:** Grafana se conecta directamente con InfluxDB, y podrá acceder sin problema a las mediciones y datos almacenados con anterioridad. Mediante dashboards interactivos Grafana mostrará una representación visual en forma de panel de control para el usuario.

Los servicios comprendidos en el sistema a parte de su intercomunicación por la red interna del Docker Compose, también se exponen puertos externos para permitir la interacción del usuario, por ejemplo.

En la captura de pantalla siguiente se podrán observar los puertos utilizados por cada uno de los componentes. Por lo general utilizaremos estos puertos para acceder a la interfaz de cada una de las herramientas.

☐	Name	Container ID	Image	Port(s)
☐ ▾ ●	project	-	-	-
☐ ●	ctGrafana	5e3b12374d79	grafana/grafana:latest	3000:3000 ↗
☐ ●	ctMinio	5c64d82933fc	minio/minio:<none>	9000:9000 ↗ 9001:9001 ↗
☐ ●	ctNifi	a21705989ae7	apache/nifi:<none>	8443:8443 ↗
☐ ●	ctInflux	d41289c1d3aa	influxdb:2.7	8086:8086 ↗

Figura 3.4. Captura con los servicios ejecutados desde Docker Desktop. Fuente: Elaboración propia

Esta separación hace que cada contenedor sea independiente pudiendo crear una comunicación protegido, pero mediante estos puertos públicos, permitirán el acceso controlado mediante credenciales de autenticación (definidas en el archivo docker-compose.yml).

Mediante estos modelos de comunicación obtenemos varias ventajas:

- **Seguridad:** Las comunicaciones internas se encuentran aisladas y no son accesibles desde el exterior.
- **Escalabilidad:** Se podrán añadir nuevos servicios o contenedores de una forma fácil y eficiente en cualquier momento, sin tener que modificar la estructura principal del sistema.
- **Automatización:** Los datos seguirán su flujo de manera totalmente automática, desde su obtención o captura hasta su visualización.

Gracias a los modelos de comunicación que se utilizan en este proyecto, se logra que cada uno de los elementos que componen el sistema actúa de manera eficiente, con coherencia y pudiendo tener los datos a disposición en todo momento. El funcionamiento de todas las piezas estará en sincronización.

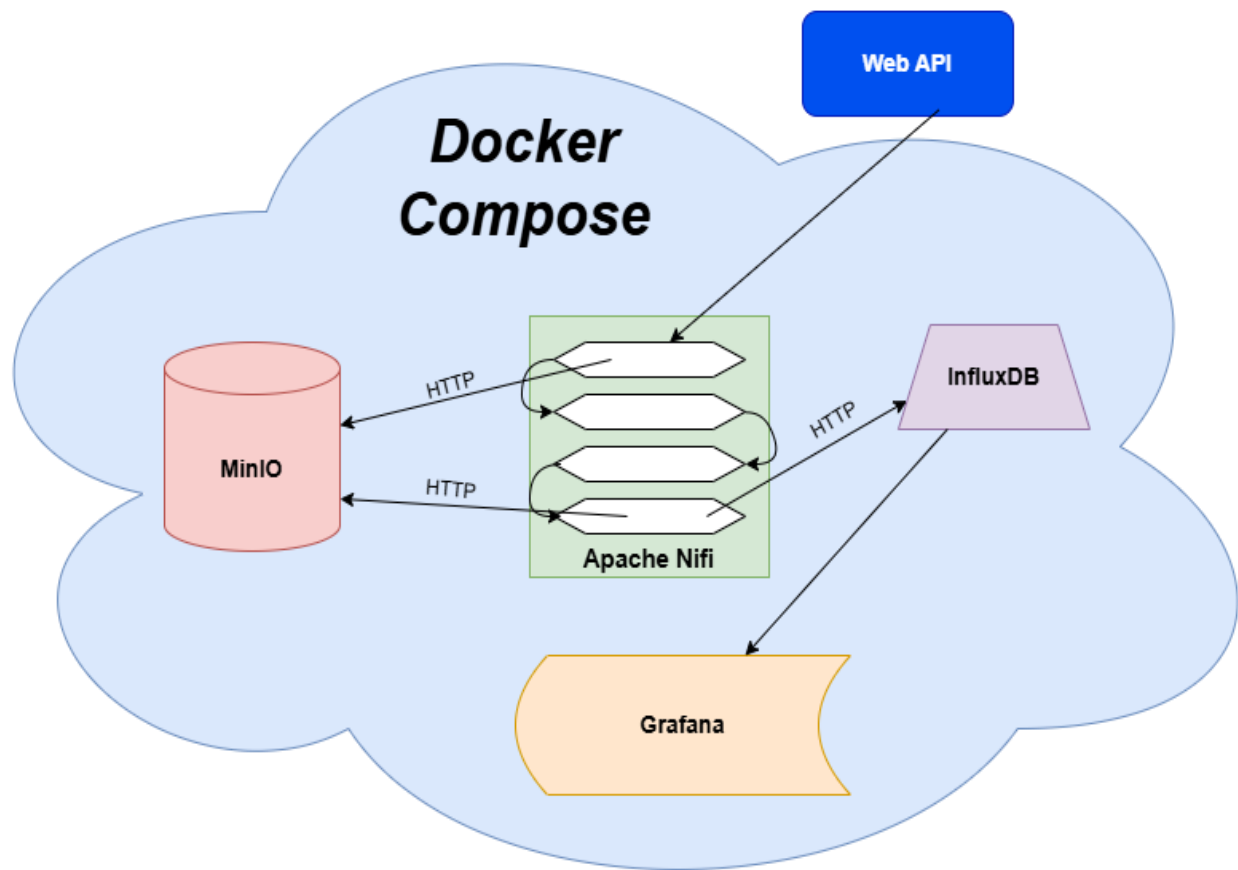


Figura 3.5. Flujo de comunicación entre servicios. Fuente: Elaboración propia

Capítulo 4

4. Análisis y diseño

En el siguiente apartado se procede a describir el trabajo de análisis y diseño del sistema. En él se examinan los datos utilizados, el modelado del dominio y se mostrará el diseño del flujo de datos.

4.1 Conocimiento de los datos

Esta sección presenta el origen, la naturaleza y las características de los datos empleados en el sistema. Se analizan las fuentes de información, estaciones de control, mediciones, estructura de los datos y los umbrales de calidad del aire existentes.

4.1.1. Fuentes de información

Los datos que recorrerán el flujo completo del sistema son captados desde las APIs correspondientes al servicio de la Junta de Castilla y León de datos de red de control de la calidad del aire [25] y de AEMET (Agencia Estatal de Meteorología) [26]. Los datos podrán ser visualizados desde las webs de cada sitio por cualquier usuario, aunque, para hacer posible la automatización será necesario el acceso a la información desde cada una de las APIs, pudiendo efectuar las solicitudes necesarias para ello. Las respuestas obtenidas desde cada API tendrán JSON como formato.

Las mediciones, no serán actualizadas en tiempo real en sus correspondientes sitios web, por lo que el sistema, irá solicitando tanto las mediciones de calidad del aire como las meteorológicas de unos días atrás. Las capturas se realizan con un cierto margen, para asegurarse de que las peticiones contengan las mediciones necesarias. En cuanto a calidad del aire, las peticiones van dirigidas a la fecha correspondiente a 4 días anteriores a la fecha actual, mientras que las realizadas sobre la API de AEMET.

Apache Nifi es el elemento encargado de efectuar las peticiones automáticamente mediante HTTP, dirigidas a las APIs de los sitios que actúan como fuentes de información para alimentar el sistema.

4.1.2 Estaciones de control

Las estaciones de control constituyen la base en la que se apoya el sistema. Son puntos físicos donde se capturan y registran las variables empleadas, temperaturas y calidad del aire, que con posterioridad serán procesadas, almacenadas y visualizadas. Estas estaciones forman parte de redes oficiales gestionadas por organismos públicos, principalmente la Agencia Estatal de Meteorología (AEMET) y los centros autonómicos de calidad del aire, dependientes en este caso de la Junta de Castilla y León.

Los datos de temperatura son provenientes de la red de observación en superficie de AEMET, constituida por estaciones meteorológicas distribuidas por todo el territorio nacional. Estos puntos pueden ser:

- ***Autonómicas:*** equipadas con sensores electrónicos que miden continuamente los valores meteorológicos y los transmiten en tiempo real.

- ***Convencionales o manuales:*** gestionados por observadores que registran las temperaturas con termómetros específicos.

Cada estación posee un identificador único y sus datos son publicados mediante la API de AEMET tras la realización de diversos controles. Se incluyen en la API todas las observaciones de todas las estaciones que se encuentren activas, entre ellas las ubicadas en Castilla y León.

Los datos de las concentraciones de contaminantes provienen de las estaciones automáticas de control de calidad del aire gestionadas por la Junta de Castilla y León. Cada estación dispone de analizadores específicos para cada contaminante concreto, por lo que habrá variabilidad en el número de contaminantes utilizados. Se registran los valores de manera continua y se enviarán a los servidores centrales del sistema de la comunidad autónoma. La información recaudada será

publicada en formato digital a través del portal de datos abiertos de la Junta, desde donde podrán ser visualizados o capturados mediante peticiones HTTP automatizadas.



Figura 4.1. Diagrama fases en la corriente de datos. Fuente: Elaboración propia

Las estaciones de control son la fuente primaria de información del sistema. Partiendo de ellas se realiza el flujo completo. Gracias a esta estructura de observación, el sistema puede ofrecer una visión integrada de la calidad del aire y de las condiciones térmicas, basada en datos oficiales y actualizados procedentes de AEMET y de la Junta de Castilla y León.

4.1.3 Formatos y estructura de datos

Los datos del proyecto se manejan en distintos formatos a lo largo del flujo de datos, adaptándose a cada fase del procesamiento. Para una eficiente y correcta automatización del sistema la estructura de los datos es fundamental. A continuación, se mostrará la estructura de los datos en cada una de las etapas del flujo, diferenciando si son meteorológicos o de calidad del aire. Se seguirá el siguiente esquema para explicar los formatos y estructuras de forma sencilla que se irán utilizando.

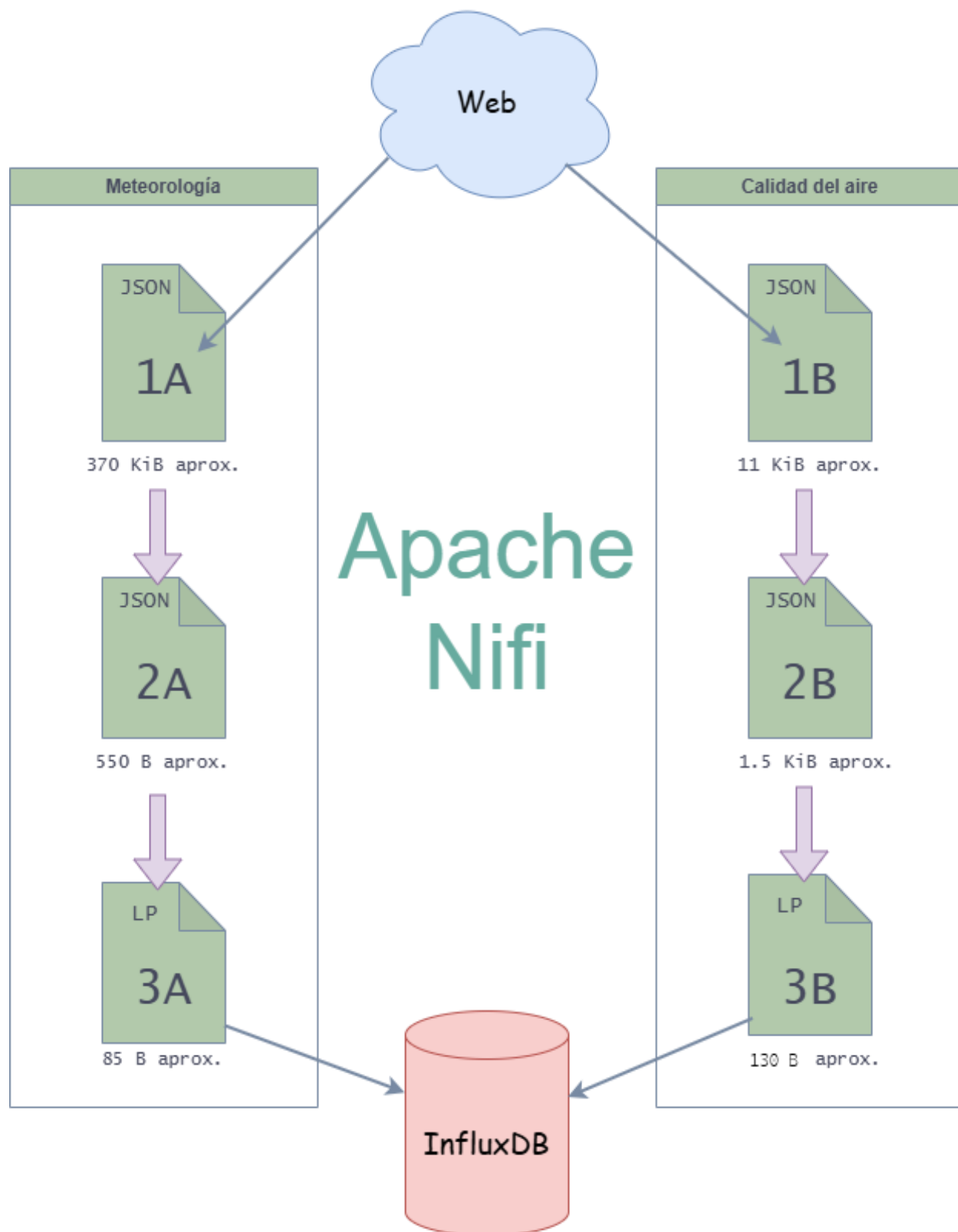


Figura 4.2. Procesos para estructura y formateo de datos. Fuente: Elaboración propia

Se utilizarán los números para indicar la etapa en la que se encuentran los archivos de datos, únicamente para simplificar la explicación. La letra A corresponderá a los archivos de datos que contienen información relativa a la meteorología y la letra B para la calidad del aire. Por ejemplo, el 2A corresponderá a la etapa dos del archivo que contiene información meteorológica. Tanto el diagrama como la propia explicación son básicos, reflejando únicamente los cambios más notables del flujo. Se va a proceder a explicar la evolución sufrida por los archivos que contienen los datos relacionados con meteorología.

- **1A:** desde la API de AEMET se extraen los datos en crudo mediante las peticiones HTTP GET correspondientes. Se obtiene un archivo JSON con toda la información meteorológica de todos los puntos de control del territorio nacional. Aproximadamente su volumen será de 370 kibibytes. A la derecha veremos un ejemplo de uno de los elementos del array que compone este archivo. Incorpora mediciones sobre viento, temperaturas, precipitaciones entre otros.

```
}, {  
  "fecha" : "2025-10-23",  
  "indicativo" : "6056X",  
  "nombre" : "SAN ROQUE ",  
  "provincia" : "CADIZ",  
  "altitud" : "1",  
  "tmed" : "24,8",  
  "prec" : "0,0",  
  "tmin" : "19,6",  
  "horatmin" : "05:25",  
  "tmax" : "29,9",  
  "horatmax" : "11:38",  
  "dir" : "28",  
  "velmedia" : "2,5",  
  "racha" : "10,3",  
  "horaracha" : "07:50",  
  "hrMedia" : "81",  
  "hrMax" : "90",  
  "horaHrMax" : "Varias",  
  "hrMin" : "50",  
  "horaHrMin" : "11:40"  
}, {
```

Figura 4.3. Ejemplo JSON fase 1A. Fuente: Elaboración propia

- **2A:** en esta etapa, automáticamente se filtran las provincias quedando únicamente las de Castilla y León, también se seleccionan los campos necesarios, en este caso la temperatura mínima y la máxima, pertenecientes a cada estación de la región. Tras ello se efectúa la media, tanto de temperaturas máximas como de mínimas de cada provincia, obteniendo un archivo JSON de unos 550 Bytes, bastante simplificado. El JSON incluirá un array con los datos de cada provincia, donde únicamente se incluirán la temperatura máxima media, la temperatura mínima media y el nombre de la provincia.

```
{
  "provincia": "SORIA",
  "avgMaxTemp": 21.1,
  "avgMinTemp": 0.7
},
{
  "provincia": "BURGOS",
  "avgMaxTemp": 20.4,
  "avgMinTemp": -0.9
},
{
  "provincia": "SALAMANCA",
  "avgMaxTemp": 22.1,
  "avgMinTemp": 0.4
},
{
  "provincia": "LEON",
  "avgMaxTemp": 16.5,
  "avgMinTemp": 2.8
},
{
  "provincia": "VALLADOLID",
  "avgMaxTemp": 21.3,
  "avgMinTemp": 1.93
}
```

Figura 4.4. Ejemplo JSON en fase 2A. Fuente: Elaboración propia

- **3A:** la finalidad de este último paso es cambiar la estructura de los datos. InfluxDB, debe de recibir los datos en formato Line Protocol. Mediante los procesadores por los que pasa el flujo, se consigue esta transformación, preparando los datos para ser enviados a la base de datos de series temporales, integrada en el sistema. A continuación, se muestra un ejemplo de una de las 0 provincias, con la estructura final en Line Protocol.



Figura 4.5. Ejemplo LineProtocol para una provincia. Fuente: Elaboración propia

A continuación, se entrará a conocer la evolución de los archivos que incluyen los datos relacionados con la calidad del aire.

- **1B:** utilizando peticiones HTTP automatizadas se capturan los datos oficiales de calidad del aire de Castilla y León. El JSON obtenido posee la información de calidad del aire de todas las estaciones de control de la comunidad. El JSON recibido contiene un array, el cual se compone de diversos elementos, cada uno referente a un punto de control determinado. En la captura siguiente se presenta un ejemplo de uno de los elementos del array, donde se pueden observar todos los campos contenidos. Aproximadamente el archivo ocupará unos 11 KiB

```
{
  "co": 0.6,
  "codProvincia": "49",
  "fecha": "01/11/2025 00:00:00",
  "idEstacion": 81,
  "no": 5.0,
  "no2": 8.0,
  "nombreEstacion": "Zamora 2",
  "nombreProvincia": "Zamora",
  "o3": 66.0,
  "pm10": 5.0,
  "pm25": 2.0,
  "pst": null,
  "sh2": null,
  "so2": 4.0,
  "timestamp": 1761951600000
}
```

Figura 4.6. Ejemplo JSON en fase 1B. Fuente: Elaboración propia

- 2B: el tamaño ocupado se reducirá considerablemente siendo de unos 1.5 kibibytes. En esta etapa del flujo nos quedaremos con los campos que realmente nos interesan, descartando 2 contaminantes que las estaciones no son capaces de medir o no son incluidos en el proceso de publicación, como son las partículas en suspensión totales, tanto sólidas como líquidas (pst) y el sulfuro de hidrógeno (SH₂). El flujo mantendrá los 7 contaminantes principales, descritos en el apartado 2.1.1, el código de la provincia y el propio nombre de la provincia. Se calcularán las medias de los valores que cada estación aportará, para capturar únicamente el valor medio por provincia, teniendo en cuenta que hay estaciones que no son capaces de capturar los datos relacionados con algún elemento contaminante determinado.

```
{
  "codProvincia": "24",
  "nombreProvincia": "León",
  "avg_co": 0.6,
  "avg_no1": 2.38,
  "avg_no2": 6.75,
  "avg_o3": 56.67,
  "avg_pm10": 7.4,
  "avg_pm25": 4.0,
  "avg_so2": 3.38
},
{
  "codProvincia": "42",
  "nombreProvincia": "Soria",
  "avg_co": null,
  "avg_no1": 8.0,
  "avg_no2": 11.0,
  "avg_o3": 61.0,
  "avg_pm10": 4.0,
  "avg_pm25": 2.0,
  "avg_so2": 1.0
},
```

Figura 4.7. Ejemplo JSON en fase 2B. Fuente: Elaboración propia

- 3B: en esta última parte del flujo se consigue la transformación del archivo de Json a formato Line Protocol, seccionado por provincias, donde cada uno ocupará alrededor de 130 Bytes. Cada provincia contará con una línea en la que aparecerán las 7 sustancias contaminantes como campos, el nombre de la provincia en cuestión, a modo de etiqueta y una marca temporal que se añadirá automáticamente, representado el día exacto en el que los datos son capturados. Seguidamente se puede examinar una de las líneas en Line Protocol, preparada para ser enviada a la base de datos de series temporales incluida en el sistema dockerizado, donde podremos percibir que en la provincia de Ávila solo se miden 5 de los contaminantes principales.

```
aire,provincia=AVILA avg_no1=2.0,avg_no2=3.0,avg_o3=65.5,avg_pm10=3.0,avg_so2=2.0 1762017948910000000
```

Figura 4.8. Ejemplo de datos en .lp en la fase 3B. Fuente: Elaboración propia

El formato y la estructura que el flujo utiliza posee múltiples ventajas entre las cuales se pueden destacar: la facilidad de un control de la trazabilidad de los objetos que incluyen los datos, permite consultas eficientes y visualizaciones dinámicas en Grafana y a demás dotan al sistema de un alto nivel de escalabilidad pudiendo, por ejemplo, extender el número de provincias o campos tratados. Apache Nifi es el encargado de garantizar la coherencia, estructura y formato durante la transformación de la información utilizada.

4.1.4 Umbrales de calidad del aire

Los umbrales de calidad del aire son los valores establecidos por organismos oficiales, que determinan cuándo una sustancia contaminante alcanza niveles que pueden ser perjudiciales para la salud o el medio ambiente. Estos valores sirven como referencia para la interpretación de las mediciones y datos recogidos por las estaciones de control.

Para la correcta definición de los umbrales, se tienen en cuenta las referencias establecidas por distintos organismos importantes:

- Organización Mundial de la Salud (OMS), que publica guías orientativas para la protección de la salud pública [27].

- Directiva 2008/50/CE del Parlamento Europeo y del Consejo, relativa a la calidad del aire y a una atmósfera más limpia [28].

- Real Decreto 102/2011, sobre la mejora de calidad del aire, que adapta la Directiva anterior complementando con las recomendaciones de la OMS para ofrecer una referencia sanitaria más precisa [29].

A continuación, se muestra una tabla que incluye los umbrales máximos que se tomarán como referencia para cada contaminante, atendiendo a los organismos anteriormente mencionados.

Los valores de umbral presentados en la tabla anterior constituyen los valores de referencia para evaluar la calidad del aire en Castilla y León. Estos límites sirven para contextualizar los niveles y concentraciones de los elementos contaminantes más importantes.

Umbral de calidad del aire para los contaminantes monitorizados

<u>Contaminante</u>	<u>Umbral máximo alerta</u>
<i>CO (Monóxido de Carbono)</i>	10mg/m ³
<i>NO (Óxido Nítrico)</i>	Sin umbral establecido (evaluación en relación con NO ₂)
<i>NO₂ (Dióxido de Nitrógeno)</i>	200 µg/m ³
<i>O₃ (Ozono Troposférico)</i>	120 µg/m ³
<i>PM₁₀ (Partículas en suspensión ≤10 µm)</i>	50 µg/m ³
<i>PM_{2.5} (Partículas en suspensión ≤2,5 µm)</i>	15 µg/m ³
<i>SO₂ (Dióxido de Azufre)</i>	125 µg/m ³

Tabla 4.1. Cuadro con los umbrales de los contaminantes. Fuente: Elaboración propia

4.2 Modelado de dominio

El objetivo es describir cómo se estructuran y relacionan las entidades principales que intervienen en el sistema. El modelado permite representar de manera conceptual como se organiza la información. Se incluyen varios elementos clave; dos diagramas entidad-relación y un diccionario de datos.

El diagrama a continuación representa las entidades, atributos y relaciones entre los distintos componentes de los datos en crudo (recién capturados desde los sitios oficiales).

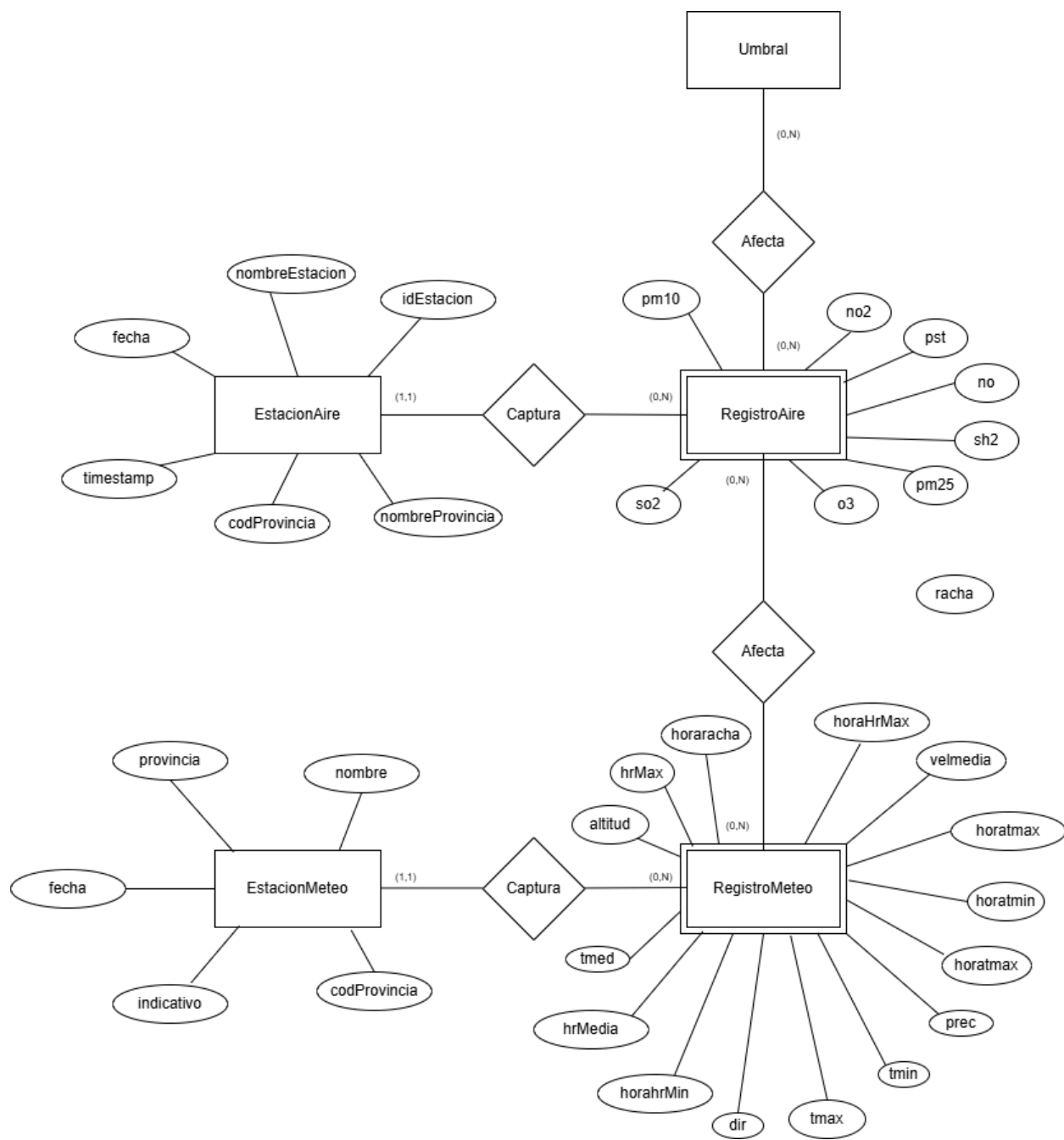


Figura 4.9. Diagrama E-R de datos en crudo. Fuente: Elaboración propia

En el siguiente modelo entidad-relación que define la organización lógica de los datos utilizados por el sistema en su fase final, tras el proceso de transformación y filtrado de la información refinada utilizada en la visualización.

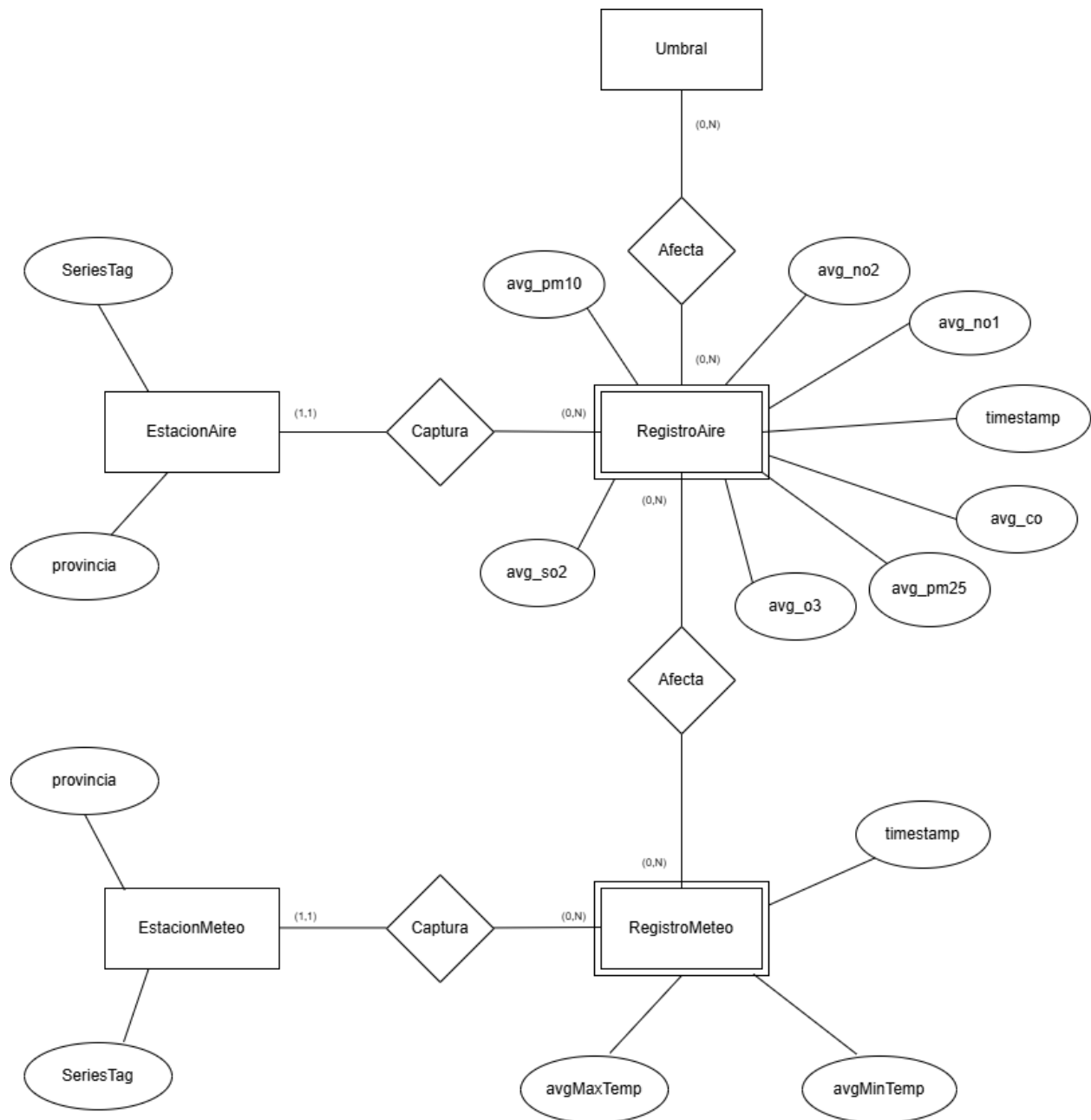


Figura 4.10. Diagrama E-R de datos procesados. Fuente: Elaboración propia

En el diccionario de datos se detallan los campos finales preparados para la visualización, garantizando la coherencia y consistencia de todos los datos gestionados por el sistema. El diccionario contiene los campos tras su procesamiento en el flujo general.

1 EstacionAire					
<u>Código</u>	<u>Atributo</u>	<u>Tipo</u>	<u>Descripción</u>	<u>Obligatorio</u>	<u>Ejemplo</u>
1.1	SeriesTag	STRING	Nombre de la serie temporal a la que pertenece	SI	aire
1.2	provincia	STRING	Nombre de la provincia	SI	SEGOVIA

Tabla 4.2. Diccionario de EstacionAire. Fuente: Elaboración propia

2 EstacionMeteo					
<u>Código</u>	<u>Atributo</u>	<u>Tipo</u>	<u>Descripción</u>	<u>Obligatorio</u>	<u>Ejemplo</u>
2.1	SeriesTag	STRING	Nombre de la serie temporal a la que pertenece	SI	temperatura
2.2	provincia	STRING	Nombre de la provincia	SI	VALLADOLID

Tabla 4.3. Diccionario de EstacionMeteo. Fuente: Elaboración propia

3 RegistroAire					
<u>Código</u>	<u>Atributo</u>	<u>Tipo</u>	<u>Descripción</u>	<u>Obligatorio</u>	<u>Unidad de medida</u>
3.1	avg_pm10	FLOAT	Valor medio de la concentración de las partículas de diámetro < a 10 µm, capturado por las estaciones de una misma provincia	NO	µg/m³
3.2	avg_no1	FLOAT	Valor medio de la concentración de monóxido de nitrógeno, capturado por las estaciones de una misma provincia	NO	µg/m³
3.3	avg_no2	FLOAT	Valor medio de la concentración de dióxido de nitrógeno, capturado por las estaciones de una misma provincia	NO	µg/m³
3.4	avg_co	FLOAT	Valor medio de la concentración de monóxido de carbono, capturado por las estaciones de una misma provincia	NO	mg/m³
3.5	avg_pm25	FLOAT	Valor medio de la concentración de las partículas de diámetro < a 2.5 µm, capturado por las estaciones de una misma provincia	NO	µg/m³
3.6	avg_o3	FLOAT	Valor medio de la concentración de ozono troposférico, capturado por las estaciones de una misma provincia	NO	µg/m³
3.7	avg_so2	FLOAT	Valor medio de la concentración de dióxido de azufre, capturado por las estaciones de una misma provincia	NO	µg/m³
3.8	timestamp	INT	Marca temporal que representa el día el que se captura el registro	SI	nanosegundos

Tabla 4.4. Diccionario de RegistroAire. Fuente: Elaboración propia

4 RegistroMeteo					
<u>Código</u>	<u>Atributo</u>	<u>Tipo</u>	<u>Descripción</u>	<u>Obligatorio</u>	<u>Unidad de medida</u>
4.1	avgMaxTemp	FLOAT	Valor medio de las temperaturas máximas recogido por todos los puntos de control de cada provincia	SI	Grados Celsius (°C)
4.2	avgMinTemp	FLOAT	Valor medio de las temperaturas mínimas recogido por todos los puntos de control de cada provincia	SI	Grados Celsius (°C)
4.3	timestamp	INT	Marca temporal que representa el día el que se captura el registro	SI	nanosegundos

Tabla 4.5. Diccionario de RegistroMeteo. Fuente: Elaboración propia

4.3 Diseño del flujo de datos

El diseño del flujo de datos establece la estructura definida que la información sigue desde su origen hasta su preparación final tras ser procesada por el sistema. En este apartado no se detallan aspectos muy técnicos ni operativos del procesamiento del sistema, si no que se describe cómo se organiza y circulan los datos de una manera general. El dataflow parte de un conjunto de mediciones e información procedente de los sitios web oficiales y los conduce a través de una serie de elementos ordenados, formando una secuencia de pasos que los archivos de información recorrerán como se puede apreciar en el diagrama siguiente:

1. La información es recibida utilizando el protocolo de comunicación HTTP, en crudo, según es publicada por los organismos emisores.

2. Se envía una copia de la información al Data Lake, para tener el archivo en crudo guardado, por seguridad y prevención.

3. Las mediciones y datos capturados se filtran y transforman siguiendo un formato coherente y simplificado que permitirá un fácil tratamiento posterior.
4. El JSON ya procesado y filtrado se guardará en nuestro servidor MinIO.
5. El sistema de procesamiento cambiará el elemento, pasando a ser Line Protocol.
6. InfluxDB recoge los archivos en formato Line Protocol.
7. Grafana captura los datos desde la base de datos InfluxDB y los muestra de una forma sencilla y ordenada, que podrá ser visualizada y utilizada por el usuario.

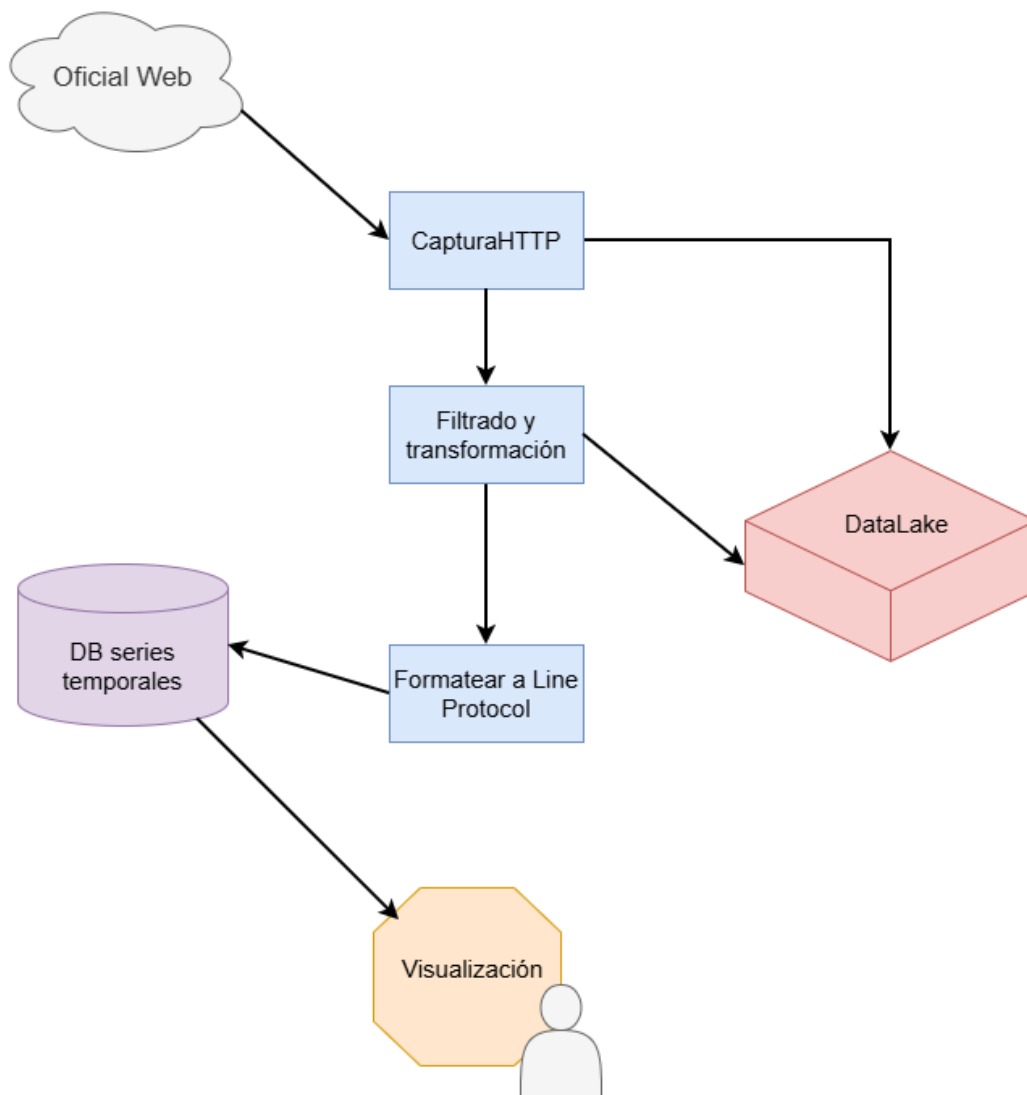


Figura 4.11. Esquema de diseño del Data-Flow. Fuente: Elaboración propia

Como se ha descrito con anterioridad los datos se capturan de varios sitios web, su estructura original es diferente y su tratamiento posterior también. El diagrama exterior refleja la secuencia de pasos utilizada, que es válida para ambos procesos, ya que aún siendo diferentes utilizan las mismas bases de procesamiento.

El diseño del data flow se fundamenta en los principios de simplicidad, claridad, y coherencia. Su objetivo principal es definir el recorrido lógico que asegure que los datos lleguen listos para su envío a la base de datos de series temporales, que actuará como fuente de datos de Grafana. En el capítulo siguiente se realizará una explicación más exhaustiva de los aspectos técnicos.

Capítulo 5

5. ETL

El sistema desarrollado se apoya en un flujo ETL (Extract, Transform, Load) encargado de la gestión automatizada del recorrido de los datos, desde su captura desde las fuentes oficiales de origen hasta su disponibilidad final para su análisis y visualización. Este proceso constituye el núcleo de operaciones del proyecto, ya que permite la organización, depuración y formateado de la información con el fin de simplificar y estar disponible para la interpretación de los servicios posteriores, como InfluxDB y Grafana.

Para una correcta comprensión del flujo, veremos los procesadores de Apache Nifi que intervienen en cada una de las fases

Flujo A. Apache Nifi GetMeteorologicalData

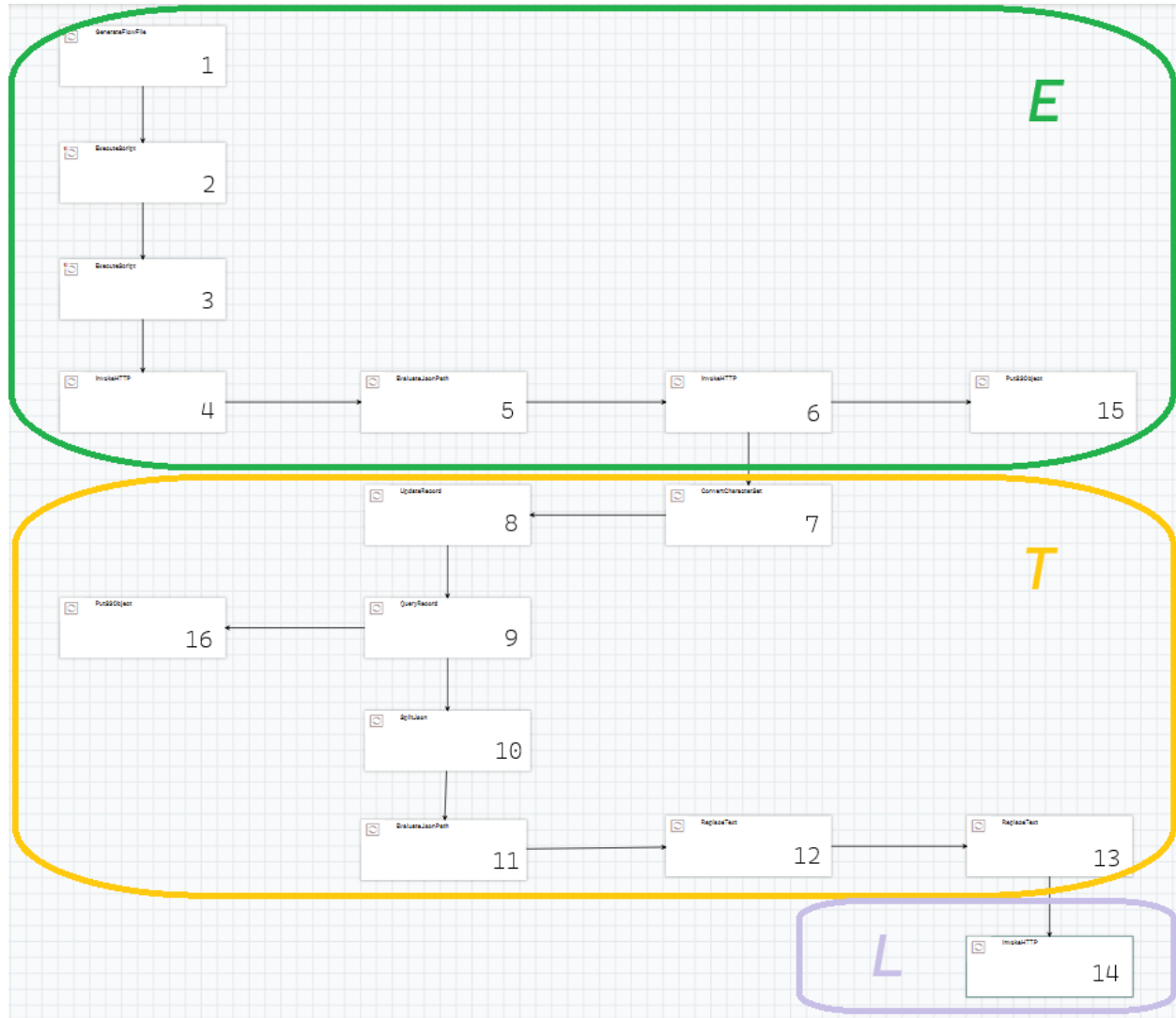
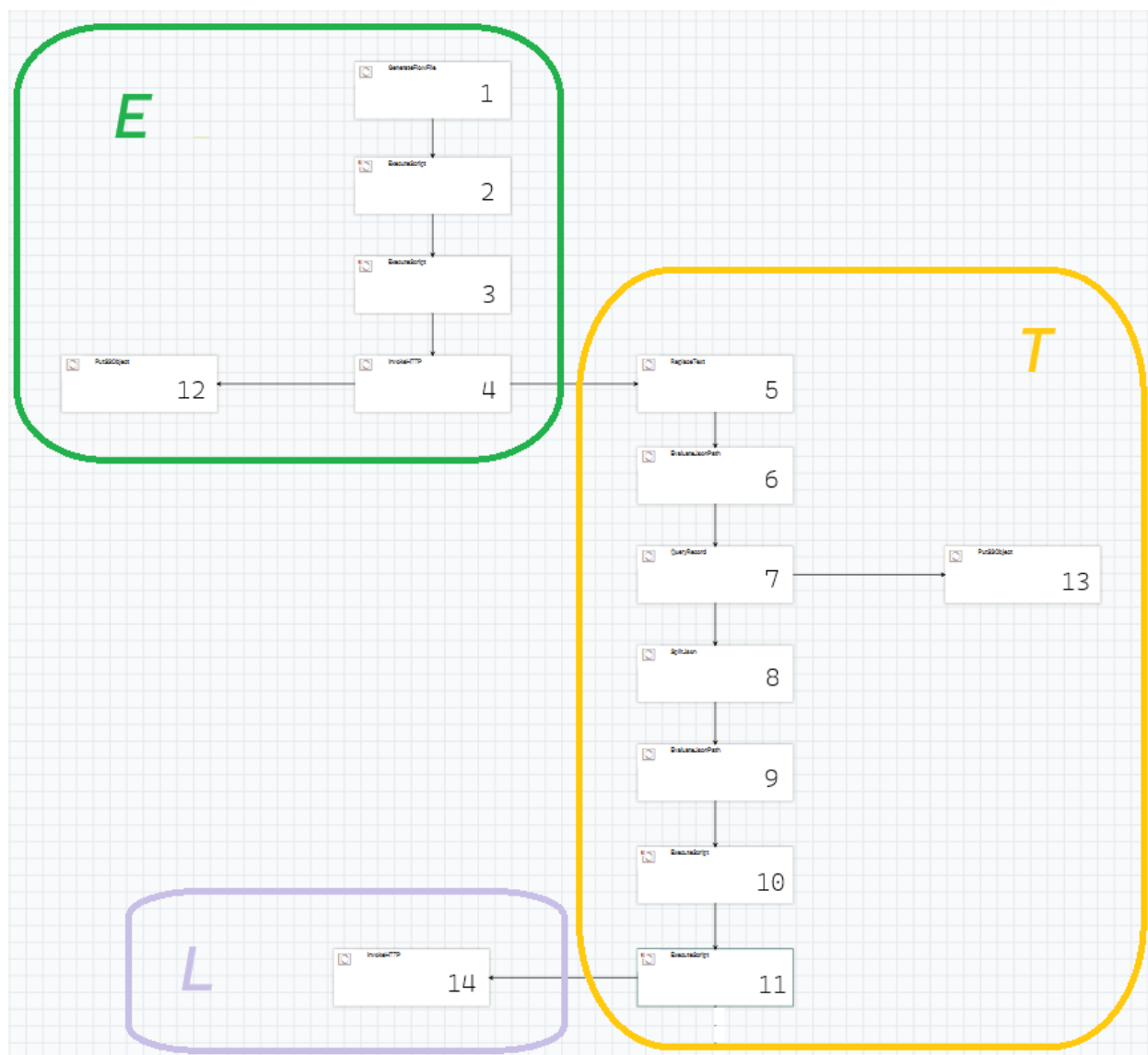


Figura 5.1. Diagrama de flujo proceso ETL datos meteorológicos. Fuente: Elaboración propia

Flujo B. Apache Nifi GetAirQualityData



*Figura 5.2. Diagrama de flujo proceso ETL datos de calidad del aire. Fuente:
Elaboración propia*

A partir de estas dos imágenes, se explicarán los pasos del proceso ETL a lo largo de este capítulo. Podemos observar ambos flujos de datos, uno para calidad del aire y otro para meteorología.

5.1 Extracción

Esta es la primera fase del proceso ETL y se centra en la obtención de los datos, tanto de calidad del aire como meteorológicos mediante solicitudes HTTP a diferentes APIs. Esta primera parte del proceso se realiza mediante Apache Nifi, actuando como motor de ingesta y responsable de establecer las conexiones con los servicios externos en todo momento.

Durante esta etapa no se realiza ninguna modificación de los datos, simplemente son recuperados desde la fuente de origen. Este paso será indispensable para que el resto del flujo se realice correctamente.

5.1.1. Origen

En este proyecto los datos provienen de dos fuentes principales:

1. API oficial de Calidad del aire de Castilla y León
2. API oficial de información meteorológica AEMET

Estas dos APIs son servicios públicos que ofrecen datos en formato JSON, sobre diferentes parámetros ambientales relevantes. Estos endpoints actúan como el punto de entrada del flujo ETL. Ambos endpoints tienen características comunes:

- Accesibles mediante HTTP, compatible con los procesadores de Nifi, permitiendo la extracción automática.
- Los datos devueltos se estructuran en formato JSON.

- La información publicada es oficial y capturada de manera fiable.
- Ambos cuentan con estabilidad continuada en el tiempo.
- Correcta y eficiente integración con un entorno dockerizado y en un flujo ETL.

En la figura 5.3 explica el ciclo natural de generación de datos antes de ser capturados por el sistema desarrollado en este proyecto.

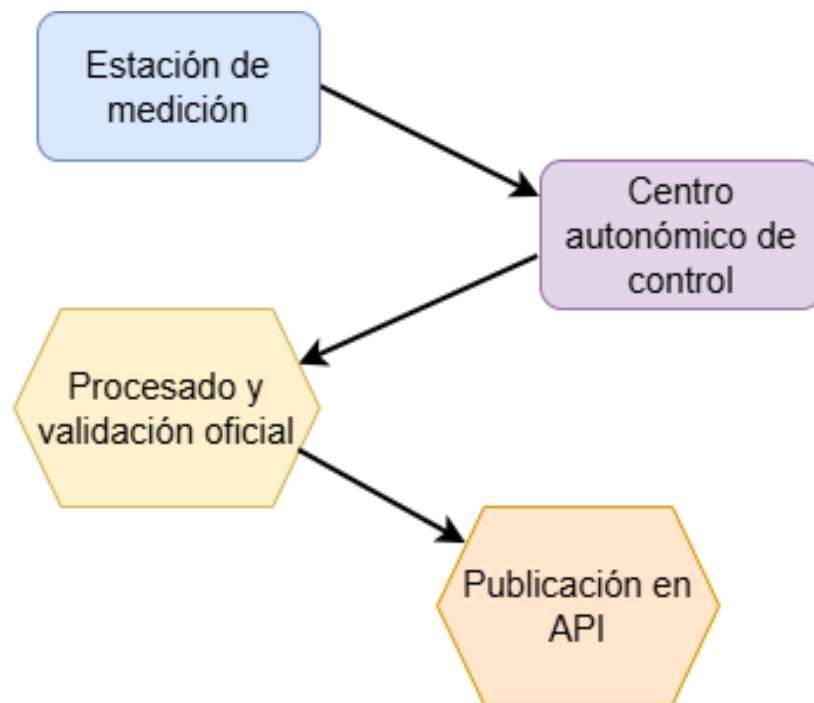


Figura 5.3. Esquema corriente de datos oficiales. Fuente: Elaboración propia

5.1.2. Ingesta

La ingesta corresponde al proceso mediante el cual los datos obtenidos desde las fuentes externas pasan a incorporarse al flujo de datos interno del sistema. La ingesta se encarga de la captura automática, periódica y controlada, de los datos publicados en las APIs oficiales.

En este sistema, la ingesta, al igual que el resto de fases del ETL, es realizada mediante Apache Nifi, actuando como motor encargado de efectuar las solicitudes a las correspondientes APIs en los intervalos temporales definidos previamente. También es el encargado de la recepción de las respuestas, en este caso en forma de fichero JSON, asegurando una correcta y eficiente captura de la información.

El objetivo principal de esta fase del data flow, es asegurar la entrada de los datos, quedando preparados para la fase de transformación, sentando las bases para siguientes pasos

Para la ingesta se utilizan los siguientes procesadores en cada flujo que quedan referenciados por número y letra siendo la letra el flujo al que pertenecen y el número, su representación en el diagrama anterior.

Flujo A. Apache Nifi GetMeteorologicalData:

- **A1.** GenerateFlowFile: inicia el flujo de trabajo, sirve como desencadenante del resto de procesadores en cadena.
- **A2.** ExecuteScript: si la hora actual coincide con la hora programada en este procesador el flujo continúa. Programado en Groovy (orientado a objetos).
- **A3.** ExecuteScript: Se crean los atributos cuyo valor correspondan a 6 días antes de la fecha actual, con el fin de prevenir los retrasos de publicación de los datos meteorológicos. Se crearán 3 atributos con la fecha en diferentes formatos, para utilizarlos a lo largo del flujo. Script elaborado en Groovy. Los atributos son los siguientes:

1. format_date.

2. format_date_url.

3. format_date_timestamp.

- **A4. InvokeHTTP:** es el encargado de realizar la solicitud HTTP a la API de AEMET utilizando el método GET. Este procesador recibe un JSON como respuesta. Se utiliza la siguiente URL para acceder a la API como cliente:

`https://opendata.aemet.es/opendata/api/valores/climatologicos/diarios/datos/fechaini/${format_date_url}/fechafin/${format_date_url}/todasestaciones`

Entre corchetes se representan las variables que se utilizarán en la URL, en este caso la fecha formateada por el procesador anterior.

Para acceder a los datos de la API de AEMET, serán necesarias 2 solicitudes HTTP, además de un `api_key` que ha sido configurada para poder ser enviada por el InvokeHTTP al efectuar la petición. La respuesta de esta solicitud será algo similar a lo siguiente.

```
{
  "descripcion" : "exito",
  "estado" : 200,
  "datos" : "https://opendata.aemet.es/opendata/sh/5faeda02",
  "metadatos" : "https://opendata.aemet.es/opendata/sh/b3aa9d28"
}
```

Figura 5.4. Respuesta JSON de la API devolviendo URL con los datos solicitados [].

Aún no se consiguen los datos correspondientes a la meteorología, la información se encuentra en la URL recibida, presente en el campo “datos” del JSON obtenido. Este funcionamiento en dos pasos es la manera estándar de acceso a datos de la API de AEMET, que utiliza URLs temporales de descarga para optimizar el rendimiento y la gestión de cargas.

- **A5. EvaluateJsonPath:** guarda la URL donde se encuentran los datos de meteorología dentro de un atributo para poder utilizarlo posteriormente.
- **A6. InvokeHTTP:** hace una request HTTP GET con la URL almacenada en el procesador anterior, esta vez sin ser necesaria la API key para recibir, ya de manera correcta, toda la información meteorológica requerida.

Flujo B. Apache Nifi GetAirQualityData

- **B1.** GenerateFlowFile: inicia el flujo de datos.
- **B2.** ExecuteScript: el flujo continuará si, y sólo si la hora programada en el Script Groovy incluido en este procesador, coincide con la hora actual. El objetivo es que el flujo se ejecute a la hora u horas, que el administrador del sistema considere.
- **B3.** ExecuteScript: en el script incluido se obtiene la fecha actual, se le restan 4 días y se generan dos versiones formateadas de la fecha en cuestión. Se añaden ambas como atributos del flujo de datos, para su utilización a lo largo del flujo. Los atributos creados son en este caso:

1. fecha_formateada.

2. fecha_formateada_minio.

El proceso de ingesta queda finalizado tras la consecución de los procesadores descritos con anterioridad.

5.1.3. Almacenamiento

Una vez ingeridos los datos desde las APIs correspondientes, Nifi los almacena directamente en MinIO, quedando una copia persistente de los datos en crudo, tal cual han sido capturados. La información en formato JSON, es enviada al datalake, a su contenedor o bucket de MinIO correspondiente. Esto permitirá un registro histórico completo, persistente en el tiempo, actuando como una copia de seguridad de fácil acceso. Según el flujo al que pertenezca la información será archivada en su contenedor o bucket correspondiente. A continuación, se muestra una captura de la interfaz principal de MinIO con los dos buckets principales.

Name	Objects	Size	Access
 air-quality-bucket	259	472.2 KiB	R/W
 meteorological-data-bucket	279	14.7 MiB	R/W

Figura 5.5. Buckets disponibles en MinIO. Fuente: Elaboración propia

Los procesadores Nifi implicados en el almacenamiento serán los siguientes:

Flujo A. Apache Nifi GetMeteorologicalData:

- **A15. PutS3Object:** este procesador envía y almacena los archivos en MinIO gracias a su compatibilidad con S3. Recibe el contenido del flujo y lo guarda en el bucket correspondiente utilizando varias configuraciones entre las cuales destacan Object Key y Bucket.

Property	Value
Object Key	 total_meteorological_data_\${format_date}_crude.json
Bucket	 meteorological-data-bucket

Figura 5.6. Propiedades del procesador PutS3Object en el flujo de meteorología.

Fuente: Elaboración propia

Flujo B. Apache Nifi GetAirQualityData

- **B12. PutS3Object:** al igual que en el flujo dedicado a la calidad del aire este procesador Nifi, tiene como única función persistir el archivo JSON recibido, garantizando que cada ejecución del flujo quede registrada como un archivo independiente dentro del data lake. Es un procesador sencillo pero fundamental para el funcionamiento del flujo y como se indicará, se utilizará con posterioridad de manera similar.

Property		Value
Object Key	?	air_quality_data_\${fecha_formateada_minio}_crude.json
Bucket	?	air-quality-bucket

Figura 5.7. Propiedades del procesador PutS3Object para calidad del aire.

Fuente: Elaboración propia

5.2 Transformación

Una vez obtenidos y guardados los datos, se procede a aplicar una serie de transformaciones, antes de enviarlos a la base de datos de series temporales para su posterior visualización. En esta fase del flujo, el objetivo principal es la adaptación de los datos contenidos en un JSON original, transformando, filtrando, adaptando y seleccionando, hasta conseguir que la información se adapte correctamente a InfluxDB.

5.2.1. Procesamiento

En esta etapa intervienen varios procesadores explicados a continuación:

Flujo A. Apache Nifi GetMeteorologicalData:

- **A7.** ConvertCharacterSet: se asegura que todos los datos utilizan una codificación estándar, en este caso UTF-8, evitando posibles errores en las etapas posteriores.
- **A8.** UpdateRecord: permite modificar los datos usando unas reglas determinadas. Se utiliza para que los valores numéricos decimales que se van a utilizar pasen de tener ‘,’ a tener ‘.’. Se evitarán problemas futuros, al efectuar este cambio en todos los valores que posean comas para diferenciar su parte decimal.
- **A9.** QueryRecord: ejecuta consultas de tipo SQL sobre los datos estructurados. Con la siguiente consulta se seleccionarán únicamente los campos del JSON referentes a las provincias que nos interesan (pertenecientes a Castilla y León). Se efectúa la media de las temperaturas máximas y mínimas de todas las estaciones de cada provincia. La consulta SQL a continuación es la incluida en este procesador Nifi, a parte del resto de configuraciones incluidas, para que el flujo pueda proseguir tras este filtrado y transformación.

```

SELECT
    provincia,
    ROUND(AVG(CAST(tmax AS DOUBLE)),2) AS avgMaxTemp,
    ROUND(AVG(CAST(tmin AS DOUBLE)),2) AS avgMinTemp
FROM FLOWFILE
WHERE provincia IN (
    'SEGOVIA',
    'PALENCIA',
    'VALLADOLID',
    'AVILA',|
    'BURGOS',
    'LEON',
    'ZAMORA',
    'SALAMANCA',
    'SORIA'
)
GROUP BY provincia

```

Figura 5.8. Consulta correspondiente al QueryRecord para la meteorología.

Fuente: Elaboración propia

El JSON obtenido es similar al capturado en el apartado 4.1.3. (Figura 4.4.)

- **A10.** SplitJson: divide el archivo en JSON más pequeños, cada uno con una sección del archivo recibido. Desde este procesador en adelante se tratará la información de cada una de las provincias de manera independiente.
- **A11.** EvaluateJsonPath: este procesador extrae valores específicos de un JSON. Mediante el uso de expresiones, se crean atributos de flujo para almacenar cada uno de los datos:
 - avgMaxTemp: media de las temperaturas máximas.
 - avgMinTemp: media de las temperaturas mínimas.
 - provincia: nombre de la provincia a la que se refieren los datos.

- **A12.** ReplaceText: reemplaza el contenido completo o parcial del archivo de flujo utilizando una plantilla o expresiones regulares. En este caso se crea una línea a partir de los atributos de flujo almacenados en el apartado anterior. Esta línea corresponderá al formato line protocol, pudiendo de esta manera ser correctamente leído por la base de datos de series temporales. El JSON obtenido es similar al capturado en el apartado 4.1.3 en la *Figura 4.8*.

En este paso, el procesador añade un timestamp correspondiente al día capturado, gracias a una correcta configuración de la expresión utilizada.

- **A13.** ReplaceText: en este procesador se efectúa un pequeño cambio, simplemente poniendo tilde a la provincia de León.

Flujo B. Apache Nifi GetAirQualityData:

- **B5.** ReplaceText: este procesador únicamente cambia el nombre del contaminante no (monóxido de nitrógeno) por no1.
- **B6.** EvaluateJsonPath: asegura que el archivo de flujo recibido por el procesador tiene formato JSON.

B7. QueryRecord: permite la ejecución de consultas SQL sobre el archivo de flujo que recibe. Nifi trata cada contenido como si de una tabla se tratara, donde cada objeto equivale a una fila. De esta manera es posible efectuar la consulta sobre un JSON. Se agrupan los registros por provincia y se efectúa la media diaria de cada contaminante redondeando a dos decimales, de cara a una posterior visualización. Se obtiene un único registro por provincia comprendiendo el valor de cada contaminante, pudiendo ser nulo, en el caso que las estaciones de la provincia no sean capaces de medir un contaminante en concreto. Al ejecutar la consulta siguiente (*Figura 5.6*) se obtendrá un JSON similar al mostrado dentro del apartado 4.1.3 (*Figura 4.7*).

La consulta utilizada para ello es la siguiente:

```

SELECT
    codProvincia,
    nombreProvincia,
    ROUND(AVG(co), 2) AS avg_co,
    ROUND(AVG(no1), 2) AS avg_no1,
    ROUND(AVG(no2), 2) AS avg_no2,
    ROUND(AVG(o3), 2) AS avg_o3,
    ROUND(AVG(pm10), 2) AS avg_pm10,
    ROUND(AVG(pm25), 2) AS avg_pm25,
    ROUND(AVG(so2), 2) AS avg_so2
FROM FLOWFILE
GROUP BY codProvincia, nombreProvincia

```

Figura 5.9. Propiedades del procesador QueryRecord para calidad del aire.

Fuente: Elaboración propia

- **B8.** SplitJson: se subdivide el JSON recibido en archivos independientes, siendo fraccionado en tantas partes como provincias comprendidas en Castilla y León.
- **B9.** EvaluateJsonPath: gracias a este procesador es posible extraer los datos del archivo de flujo recibido y almacenarlos de manera ordenada en los siguientes atributos de flujo:
 - avg_co.
 - avg_no1.
 - avg_no2.
 - avg_o3.
 - avg_pm10.
 - avg_pm25.
 - avg_so2.
 - nombreProvincia.

- **B10.** ExecuteScript: define, crea y agrega al flujo un nuevo atributo de flujo, correspondiente a la marca de tiempo, que servirá para ubicar temporalmente los archivos line protocol que se cargarán en InfluxDB. La programación de este procesador está elaborada en Groovy.

```
import java.time.ZonedDateTime

def flowFile = session.get()
if (!flowFile) return

// Calcula la marca de tiempo: hace 4 días en nanosegundos
def timestamp = ZonedDateTime.now().minusDays(4).toInstant().toEpochMilli() * 1_000_000L

flowFile = session.putAttribute(flowFile, "timestamp_line_protocol", timestamp.toString())
session.transfer(flowFile, REL_SUCCESS)
```

Figura 5.10. Script utilizado en el procesador B10. Fuente: Elaboración propia

- **B11.** ExecuteScript: en este procesador se estructuran todos los atributos de flujo capturados en los procesadores anteriores, formando una estructura line protocol. Podemos observar un ejemplo del resultado final en la *Figura 4.8* del apartado 4.1.3. El procesador ejecuta el siguiente script en Groovy.

```
import org.apache.nifi.processor.io.StreamCallback
import java.nio.charset.StandardCharsets

flowFile = session.get()
if(!flowFile) return

// Leemos atributos desde NiFi
def provincia = flowFile.getAttribute("nombreProvincia")?.toUpperCase()
def avg_co = flowFile.getAttribute("avg_co") ?: ""
def avg_no1 = flowFile.getAttribute("avg_no1") ?: ""
def avg_no2 = flowFile.getAttribute("avg_no2") ?: ""
def avg_o3 = flowFile.getAttribute("avg_o3") ?: ""
def avg_pm10 = flowFile.getAttribute("avg_pm10") ?: ""
def avg_pm25 = flowFile.getAttribute("avg_pm25") ?: ""
def avg_so2 = flowFile.getAttribute("avg_so2") ?: ""
def timestamp = flowFile.getAttribute("timestamp_line_protocol") ?: System.currentTimeMillis() * 1000000L

// Montamos la línea en Line Protocol, solo poniendo los campos que tengan valor
def fields = []
if(avg_co) fields << "avg_co=${avg_co}"
if(avg_no1) fields << "avg_no1=${avg_no1}"
if(avg_no2) fields << "avg_no2=${avg_no2}"
if(avg_o3) fields << "avg_o3=${avg_o3}"
if(avg_pm10) fields << "avg_pm10=${avg_pm10}"
if(avg_pm25) fields << "avg_pm25=${avg_pm25}"
if(avg_so2) fields << "avg_so2=${avg_so2}"

def lineProtocol = "aire,provincia=${provincia} ${fields.join(",")} ${timestamp}"

// Escribimos el contenido nuevo en el flowfile
flowFile = session.write(flowFile, { inputStream, outputStream ->
    outputStream.write(lineProtocol.getBytes(StandardCharsets.UTF_8))
} as StreamCallback)

// Pasamos el FlowFile al success
session.transfer(flowFile, REL_SUCCESS)
```

Figura 5.11. Propiedades del procesador PutS3Object. Fuente: Elaboración propia

5.2.1. Persistencia

Una vez procesada la información, tras haberse sometido a diversos procedimientos con el objetivo de preparar los datos, el archivo de flujo es guardado en el datalake, justo antes del cambio de formato a line protocol, siendo archivado como JSON.

Flujo A. Apache Nifi GetMeteorologicalData:

- **A15.** PutS3Object: este procesador envía el archivo con la información meteorológica una vez procesado y en formato JSON al contenedor correspondiente de minIO.

Property		Value
Object Key	?	total_meteorological_data_\${format_date}.json
Bucket	?	meteorological-data-bucket

Figura 5.12. Propiedades del procesador PutS3Object para la meteorología.

Fuente: Elaboración propia

Flujo B. Apache Nifi GetAirQualityData:

- **B13.** PutS3Object: de la misma manera que en el procesador anterior, se ingresan los datos referentes a la calidad del aire en Castilla y León en el datalake. En la captura a continuación se muestran dos de las propiedades configuradas en este paso.

Property		Value
Object Key	?	air_quality_data_\${fecha_formateada_minio}.json
Bucket	?	air-quality-bucket

Figura 5.13. Propiedades del procesador PutS3Object para la meteorología.

Fuente: Elaboración propia

5.3 Carga

La fase de carga representa el último de los pasos dentro del proceso de ETL y es la responsable de poner a disposición del sistema final los datos ya procesados. Tras haber sido extraídos, filtrados y transformados, la información resultante debe de ser enviada y almacenada en una plataforma preparada para su posterior consulta, análisis o visualización.

En esta fase, el sistema asegura que los datos lleguen de manera íntegra y en formato adecuado. La carga comprende tanto el proceso de transmisión de la información al hasta el lugar de carga, hasta su posterior consumo, en este caso, desde la herramienta de visualización.

5.3.1. Transferencia de datos

En esta subfase los datos pasan se guardan en la base de datos preparada para series temporales que el sistema utilizará. La información ya preparada y lista, pasará desde Apache Nifi hasta InfluxDB utilizando el protocolo HTTP para la transferencia. Existirá un archivo para cada una de las 9 provincias de Castilla y León con la información concreta de cada una, ya sea referente a la calidad del aire o a la meteorología.

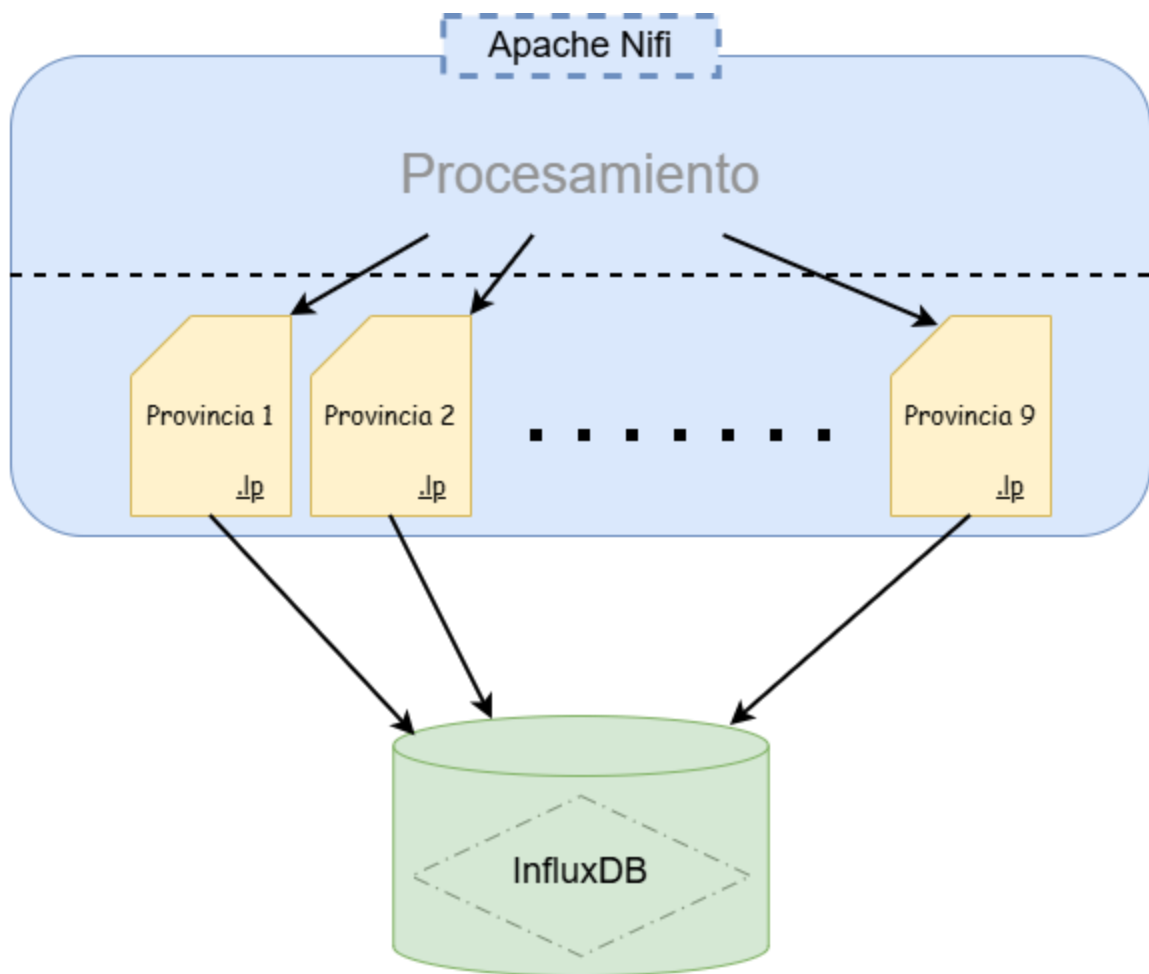


Figura 5.14. Diagrama explicativo del proceso de transferencia hacia InfluxDB.

Fuente: Elaboración propia

A continuación, se mostrarán los procesadores responsables de esta transferencia de los datos hasta InfluxDB. Los datos a transferir se encuentran en formato line protocol, ejemplo en *Figura 4.5* y *Figura 4.8* del apartado 4.1.3, teniendo ya la estructura final para ser enviados a la base de datos de series temporales.

Flujo A. Apache Nifi GetMeteorologicalData:

- **A14.** InvokeHTTP: utilizado para el envío de las líneas line protocol generadas en el flujo hacia el punto de entrada de la base de datos. Una vez que el procesador recibe el contenido ya preparado, efectúa una petición HTTP al servicio de destino y se hace una entrega de datos sin más pasos intermedios. Se necesitará un token de autorización generado por InfluxDB para que esta comunicación sea posible.

Flujo B. Apache Nifi GetAirQualityData:

- **B14.** InvokeHTTP: al igual que en el flujo de meteorología este procesador de Nifi hace posible la transferencia entre servicios de los datos de calidad del aire, mediante HTTP POST. Es necesario el token de autorización para que la información sea correctamente enviada mediante Apache Nifi y recibida por InfluxDB.

5.3.2. Consumo de datos

Una vez que los datos se encuentran en InfluxDB comienza la etapa de consumo. Una vez que la información ya se encuentra archivada y organizada, y se encuentra a disposición del servicio que accederá a ella para ser consultada con el objetivo de hacer análisis o visualizarla. En este sistema se utilizará Grafana, que será la herramienta que representa el último eslabón de la cadena y enlace con el usuario final.

En este sistema de almacenamiento (InfluxDB) los datos se organizan como series temporales, es decir, cada registro se encuentra asociado a una marca de tiempo que actúa como clave principal. Cada uno de estos registros o mediciones contiene etiquetas (tags) y un conjunto de campos (fields) que contienen los valores medidos de forma numérica o textual. Gracias a este tipo de estructura, tanto la inserción como la consulta es óptima, ya que se indexa basándose en los tags para localizar de manera rápida las series requeridas. Influx usa también un motor interno orientado a compresión para almacenar gran cantidad de datos con un buen rendimiento.

Grafana consulta los datos directamente mediante el conector de series temporales, que envía solicitudes parametrizadas utilizando, por ejemplo, rangos temporales o filtrado por etiquetas. El sistema internamente traduce estas peticiones al lenguaje propio del motor de Influx, devolviendo únicamente los datos necesarios para la representación de paneles, gráficas o tablas. Grafana no transforma ni almacena los datos, únicamente interpreta la respuesta, la organiza y estructura de forma visual según la configuración establecida con anterioridad.

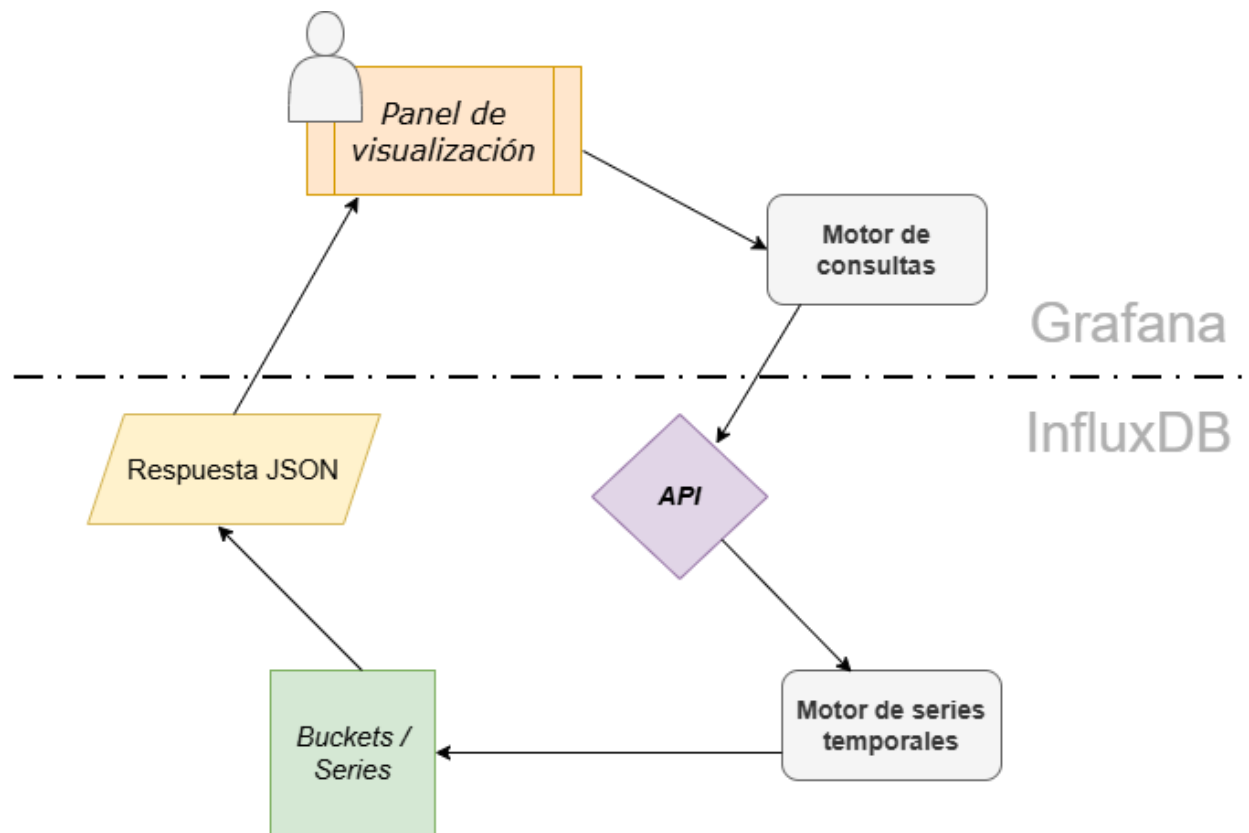


Figura 5.15. Esquema de interacciones Grafana-InfluxDB. Fuente: Elaboración propia

En resumen, el consumo de datos tiene como objetivo que la información almacenada en el sistema de series temporales pueda ser accedida de manera sencilla y rápida. Gracias a ello la capa de visualización obtiene los datos ya preparados y pueden ser representados sin tener que hacer transformaciones adicionales.

Capítulo 6

6. Exploración

6.1 Planificación

6.1.1. Propósito

En este trabajo se desarrolla un sistema completo para la obtención, almacenamiento y visualización de datos ambientales, utilizando tecnologías cuya disposición se encuentra basada en contenedores. A partir de esta infraestructura, resulta necesario definir cómo se van a presentar los datos capturados y transformados para poder ser comprendidos y analizados de forma efectiva.

A continuación, se exponen los objetivos de la fase de exploración del proyecto:

- La visualización debe de estar preparada para utilizarse como medio para el análisis de los datos ambientales generados por el sistema, permitiendo la conversión de los valores y mediciones en representaciones que faciliten su interpretación dentro del contexto del proyecto
- Estudiar la evolución temporal de las distintas sustancias contaminantes y variables meteorológicas disponibles, con el fin de identificar tendencias, variaciones relevantes y posibles comportamientos anómalos en los datos registrados.
- Comprobar la coherencia y consistencia de los datos a lo largo del flujo completo, utilizando los paneles de visualización como herramienta de verificación del correcto funcionamiento del sistema desde la ingesta hasta su presentación final.
- Presentación de la información de una manera clara y accesible, para poder ser fácilmente comprendida y utilizada sin necesidad de conocimientos técnicos avanzados, cumpliendo uno de los objetivos principales del sistema.

En conjunto, los propósitos expuestos delimitan la orientación del apartado y aseguran que la visualización responda a las necesidades del proyecto.

6.1.2. Enfoque

La exploración de los datos se aborda desde una perspectiva práctica, partiendo desde la información ya almacenada y estructurada para su correcto análisis visual. El enfoque no se centra en la generación de nuevas mediciones, sino en la observación y comprensión de los datos ya disponibles, mediante la representación de gráficos dinámicos y configurables.

Se opta por un enfoque basado en series temporales, en el que el tiempo actúa como eje principal del estudio permitiendo examinar la evolución de los valores registrados y su comportamiento a lo largo de distintos periodos. Este planteamiento facilita la identificación de posibles anomalías, cambios o patrones, sin la necesidad de realizar transformaciones adicionales sobre los datos.

En este sistema, la presentación se apoya en la parametrización de las visualizaciones, permitiendo ajustar tanto los rangos temporales como la ubicación en la que las mediciones han sido tomadas.

El enfoque correspondiente a esta fase, como se ha explicado con anterioridad, prioriza la interpretación directa de los datos mediante herramientas visuales, manteniendo una separación clara entre las fases de tratamiento de la información y su posterior exploración.

6.1.3. Diseño

En este apartado se concretan los elementos que intervienen en la construcción de las visualizaciones, definiendo tanto las variables utilizadas como el tipo de presentación adoptada, en coherencia con los objetivos y pautas marcadas en los apartados anteriores.

La presentación de los datos se basa en los datos ya procesados y disponibles en el sistema, considerando las siguientes variables:

- Mediciones de calidad del aire.
- Valores de temperatura.
- Información temporal asociada a cada medición.
- Identificación territorial por provincias.

Para la presentación de estas variables se opta por una visualización tipo dashboard, concebida como un cuadro de mando que permite centralizar la información. Este tipo de presentación prioriza mostrar una versión global clara, reduciendo la complejidad visual y facilitando la interpretación de los datos de un solo vistazo.

6.2 Implementación

En este apartado se describe la materialización de la fase de exploración, detallando cómo se han construido las visualizaciones, como se presentan y cómo el usuario podrá interactuar con ellas.

6.2.1. Construcción de las visualizaciones

A continuación, se explicará el proceso de creación y organización estructural de las diferentes representaciones finales, elaboradas a partir de los datos ya almacenados por el sistema. Estos datos se encuentran organizados en la base de datos de series temporales, a la cual se accede automáticamente mediante un lenguaje de consultas llamado InfluxQL (basado en SQL). Este lenguaje es utilizado debido a su compatibilidad con InfluxDB. Con él se permite que el acceso a los datos desde la herramienta de presentación (Grafana), se realice de una forma eficiente y rápida.

En primer lugar, se crea un dashboard en el que se encuentran varios apartados. Cada uno de ellos corresponde a una medición diferente, siendo estos apartados desplegados, pudiendo ocultar la información comprendida dentro de cada uno independientemente. En la imagen a continuación se presentan los diferentes apartados desplegados contenidos en el dashboard.



Figura 6.1. Grafana, apartados desplegados del dashboard. Fuente: Elaboración propia

A su vez, dentro de cada apartado, aparece toda la información referente a su título organizada mediante paneles de diferentes tipos, para poder lograr una presentación de los datos clara, concisa y muy estructurada pudiendo, de un solo vistazo, localizar lo que el usuario decida.

Cada uno de los paneles está diseñado en base a dos variables, que podrán ser modificadas por el usuario. Tras el cambio de las variables, los paneles de información comprendidos en todo el dashboard, se refrescarán inmediatamente atendiendo a la selección de usuario. Los valores seleccionables son:

- Provincia: Aparecerán las 9 provincias de Castilla y León.
- Rango de tiempo: Se podrá seleccionar cualquier rango de tiempo.

Para la creación de los paneles es necesario

6.2.2. Detalle de las visualizaciones

Se describen en esta parte las distintas visualizaciones desarrolladas para la exploración de datos, atendiendo a la naturaleza de la información representada y los objetivos de análisis planteados. Para facilitar su comprensión, se explicarán dentro de diferentes apartados correspondientes, según el tipo de información contenida.

Se presentará el tipo de paneles utilizados, la unidad de medida utilizada, detalles propios, códigos utilizados para su montaje y otros aspectos a tener en cuenta. El proceso de creación de cada panel va más allá del código utilizado (presentado en las tablas analíticas de cada panel posteriormente), siendo necesarios muchos ajustes y transformaciones adicionales.

6.2.2.1 Visualizaciones de temperatura

En este subapartado se incluyen los paneles centrados en las mediciones de temperatura registradas, con el objetivo de analizar su evolución temporal y su comportamiento a lo largo de diferentes periodos considerados. Se muestra seguidamente una captura de pantalla correspondiente al apartado desplegable.

Temperaturas



Figura 6.2. Grafana, apartado temperaturas. Fuente: Elaboración propia

En este apartado se cuenta, cómo se puede observar, con 5 paneles diferentes. Se hará referencia a ellos como “Panel de Temperaturas” y el número correspondiente (PTx) como identificador de cada uno:

PT1	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Grados centígrados (C°)
Descripción	Muestra el valor máximo de las temperaturas registradas en el lugar y periodo de tiempo seleccionados. Aparecerá en color naranja.
Código de creación	<code>SELECT max("avgMaxTemp") FROM "temperatura" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.1. Panel, valor máximo de temperaturas. Fuente: Elaboración propia

PT2	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Grados centígrados (C°)
Descripción	Muestra el valor mínimo de las temperaturas registradas. Aparecerá el valor en color azul.
Código de creación	SELECT min("avgMaxTemp") FROM "temperatura" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter

Tabla 6.2. Panel, valor mínimo de temperaturas. Fuente: Elaboración propia

PT3	PROMEDIO DE T° MÁXIMAS
Tipo de panel	Stat
Unidad de medida	Grados centígrados (C°)
Descripción	Muestra el promedio de todas las temperaturas máximas disponibles en el lugar y tiempo seleccionado. Aparecerá el valor en verde.
Código de creación	SELECT mean("avgMaxTemp") FROM "temperatura" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter

Tabla 6.3. Panel, promedio de temperaturas máximas. Fuente: Elaboración propia

PT4	PROMEDIO DE T° MÍNIMAS
Tipo de panel	Stat
Unidad de medida	Grados centígrados (C°)
Descripción	Muestra el promedio de todas las temperaturas mínimas registradas. El valor se representa en verde.
Código de creación	SELECT mean("avgMinTemp") FROM "temperatura" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter

Tabla 6.4. Panel, promedio de temperaturas mínimas. Fuente: Elaboración propia

6.2.2.2 Visualizaciones de calidad del aire

Se presentan en este grupo, las visualizaciones centradas exclusivamente en los contaminantes atmosféricos monitorizados, permitiendo observar sus niveles, variaciones y posibles episodios relevantes en función del tiempo y su situación geográfica. A continuación, se explicarán las presentaciones de los valores correspondientes a los 7 contaminantes registrados.

Los tres paneles de la zona izquierda serán comunes para todos, mostrando el valor máximo, el mínimo y el promedio de todos los valores captados sobre el contaminante en cuestión. Se mostrará con posterioridad cada apartado desplegable del dashboard, incluyendo los paneles dedicados correspondientes. Todos tendrán la misma estructura con 4 paneles en una misma disposición. El identificador de cada panel de calidad del aire será formado como “PCAx”, donde x es el número de panel. Para una comprensión mayor de estos apartados correspondientes a la calidad del aire, ver la sección 2.2.1 Elementos contaminantes, en este documento.

Valores de NO2 (Dióxido de nitrógeno)



Figura 6.3. Grafana, apartado NO2. Fuente: Elaboración propia

En la captura anterior se muestra la estructura que tienen todos los apartados que contienen la información de las sustancias contaminantes del aire.

Los paneles que componen el apartado dedicado al NO2 son:

PCA1	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el máximo valor registrado de NO2 en el lugar y periodo de tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_no2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.5. Panel, valor máximo NO2. Fuente: Elaboración propia

PCA2	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el mínimo valor registrado de NO2. Se muestra en color azul.
Código de creación	<code>SELECT min("avg_no2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.6. Panel, valor mínimo NO2. Fuente: Elaboración propia

PCA3	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra la media de todos los valores almacenados en la localización geográfica y tiempo seleccionados del contaminante NO2. Se muestra en verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_no2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.7. Panel, valor promedio NO2s. Fuente: Elaboración propia

PCA4	GRÁFICO VALORES NO2
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Gráfico con los valores de NO2. En el eje "x" aparece el rango de tiempo seleccionado mientras que en el eje "y" la concentración de dióxido de nitrógeno. En la parte superior derecha está la leyenda, comprendiendo 3 variables, los cuales son, valores de NO2, promedio móvil y el umbral. Se podrán mostrar independientemente o juntos, según el usuario desee.
Código de creación	<code>SELECT last("avg_no2") AS "NO2 diario" FROM "aire"</code> <code>WHERE ("provincia"::tag =~ /^\$provincia\$/) AND</code> <code>\$timeFilter GROUP BY time(1d) fill(none)</code> <code>SELECT moving_average(mean("avg_no2"), 5) FROM "aire"</code> <code>WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code> <code>GROUP BY time(1d) fill(null)</code>

Tabla 6.8. Panel, gráfico valores NO2. Fuente: Elaboración propia

Valores de NO (Monóxido de nitrógeno)

A continuación, se mostrará el panel que contiene el gráfico correspondiente al contaminante NO, teniendo en cuenta que tiene a su izquierda otros 3 paneles representando el máximo, el mínimo y el promedio.

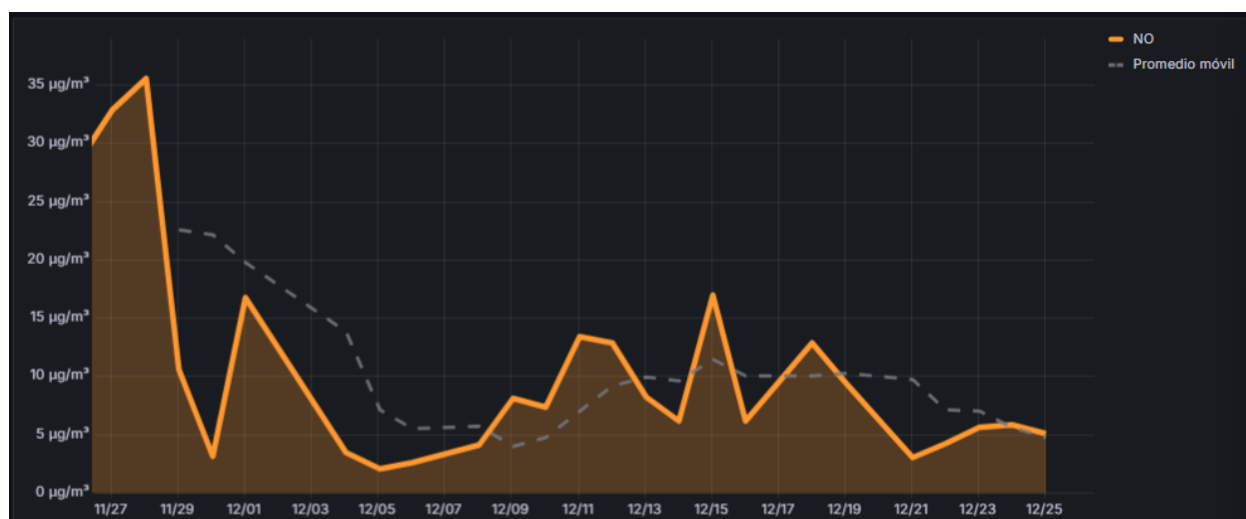


Figura 6.4. Grafana, apartado NO. Fuente: Elaboración propia

Se presentan a continuación las tablas explicativas correspondientes al NO:

PCA5	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el máximo registrado de NO en el lugar y periodo de tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_no") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.9. Panel, valor máximo NO. Fuente: Elaboración propia

PCA6	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor mínimo registrado de NO. Aparece en azul.
Código de creación	<code>SELECT min("avg_no") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.10. Panel, valor mínimo NO. Fuente: Elaboración propia

PCA7	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Representa el promedio de todos los valores almacenados en el lugar y tiempo señalados de monóxido de nitrógeno registrado. Aparece en verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_no") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.11. Panel, valor promedio NO. Fuente: Elaboración propia

PCA8	GRÁFICO VALORES NO
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Gráfico con los valores de NO. En el eje “x” aparece el tiempo seleccionado mientras que en el eje “y” la concentración de monóxido de nitrógeno. En la parte superior derecha está la leyenda, comprendiendo 2 variables, las cuales son, valores de NO y promedio móvil careciendo en este caso de umbral.
Código de creación	<pre>SELECT last("avg_no1") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none) SELECT moving_average(mean("avg_no1"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</pre>

Tabla 6.12. Panel, gráfico con valores NO. Fuente: Elaboración propia

Valores de SO2 (Dióxido de azufre)

Acto seguido encontramos una captura del apartado dedicado al dióxido de azufre (SO2) y las tablas con la información dedicada a los paneles incluidos en este apartado. Los 3 paneles de la izquierda indican y poseen la misma estructura que los apartados anteriores.

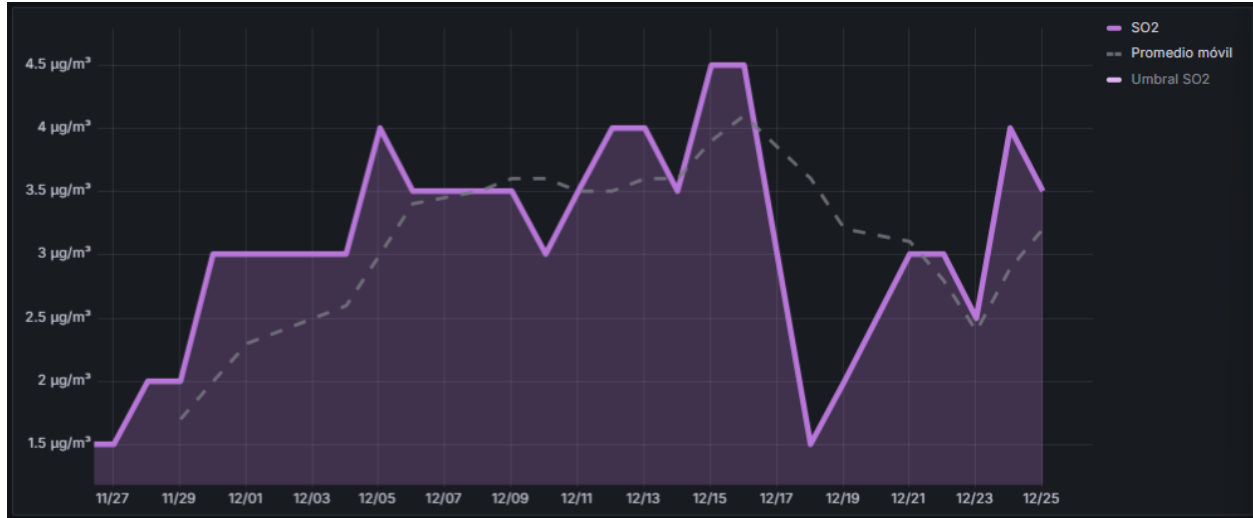


Figura 6.5. Grafana, apartado SO2. Fuente: Elaboración propia

PCA9	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el máximo registrado de SO2 en el lugar y periodo de tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_so2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.13. Panel, valor máximo SO2. Fuente: Elaboración propia

PCA10	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor mínimo registrado de SO2 en el lugar y tiempo correspondientes. Aparece en color azul.
Código de creación	<code>SELECT min("avg_so2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.14. Panel, valor mínimo SO2. Fuente: Elaboración propia

PCA11	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Representa el promedio de todos los valores almacenados en el lugar y tiempo señalados de monóxido de nitrógeno registrado. Aparece en verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_so2") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.15. Panel, valor promedio de SO2. Fuente: Elaboración propia

PCA12	GRÁFICO VALORES SO2
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Gráfico con los valores de SO2. En el eje "x" aparece el tiempo mientras que en el eje "y" la concentración de dióxido de azufre. En la parte superior derecha está la leyenda, comprendiendo 3 variables, las cuales son, valores de SO2, promedio móvil y umbral.
Código de creación	<pre>SELECT last("avg_so2") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none) SELECT moving_average(mean("avg_so2"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</pre>

Tabla 6.16. Panel, gráfico valores SO2. Fuente: Elaboración propia

Valores de CO (Monóxido de carbono)

Seguidamente encontraremos una captura del apartado desplegable dedicado al monóxido de carbono y las figuras con la información sobre los paneles que lo componen.

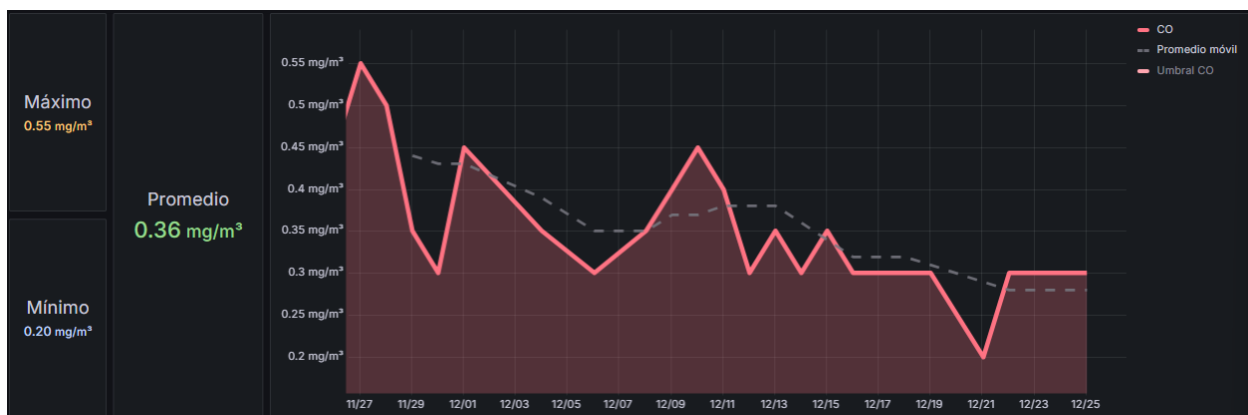


Figura 6.6. Grafana, apartado CO. Fuente: Elaboración propia

PCA13	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Miligramos por metro cúbico (mg/m ³)
Descripción	Muestra el valor máximo de CO en la localización y periodo de tiempo seleccionados. Se muestra en naranja.
Código de creación	<code>SELECT max("avg_co") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.17. Panel, valor máximo CO. Fuente: Elaboración propia

PCA14	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Miligramos por metro cúbico (mg/m ³)
Descripción	Muestra el valor mínimo registrado de CO en el lugar y tiempo correspondientes. Aparece en color azul.
Código de creación	<code>SELECT min("avg_co") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.18. Panel, valor mínimo CO. Fuente: Elaboración propia

PCA15	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Miligramos por metro cúbico (mg/m ³)
Descripción	Representa el promedio de todos los valores almacenados en el lugar y tiempo señalados de monóxido de nitrógeno registrado. Aparece en verde y a la derecha de los dos paneles anteriores.
Código de creación	SELECT mean("avg_co") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter

Tabla 6.19. Panel, valor promedio CO. Fuente: Elaboración propia

PCA16	GRÁFICO VALORES CO
Tipo de panel	Time Series
Unidad de medida	Miligramos por metro cúbico (mg/m ³)
Descripción	Gráfico dedicado al Monóxido de carbono. En el eje "x" se encuentra el tiempo mientras que en el eje "y" la concentración de CO. En la parte superior derecha está la leyenda, comprendiendo los valores del contaminante, el promedio móvil y el umbral, pudiendo cambiar de manera dinámica entre ellas.
Código de creación	<pre>SELECT last("avg_co") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none) SELECT moving_average(mean("avg_co"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</pre>

Tabla 6.20. Panel, gráfico valores CO. Fuente: Elaboración propia

Valores de PM2.5 (Partículas en suspensión)

Se presenta una captura del gráfico correspondiente al apartado sobre las partículas en suspensión con un diámetro de 2.5 micrómetros o menos. Veremos en la imagen que se encuentra visible el umbral, disponible para su visualización en la leyenda del panel. Los paneles que contienen el valor máximo, mínimo y promedio estarán estructurados de la misma forma que los anteriores.



Figura 6.7. Grafana, apartado PM2.5. Fuente: Elaboración propia

PCA17	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor máximo de partículas en suspensión cuyo diámetro es menor a 2.5 micrómetros en la localización y periodo de tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_pm25") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.21. Panel, valor máximo PM25. Fuente: Elaboración propia

PCA18	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor mínimo registrado de PM2.5 en el lugar y tiempo correspondientes. Aparece en azul.
Código de creación	<code>SELECT min("avg_pm25") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.22. Panel, valor mínimo PM25. Fuente: Elaboración propia

PCA19	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Representa el valor promedio de todos los valores almacenados en el lugar y tiempo señalados de las concentraciones de PM2.5 registradas. Aparece en color verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_pm25") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.23. Panel, valor promedio PM25. Fuente: Elaboración propia

PCA20	GRÁFICO VALORES PM2.5
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Gráfico dedicado a PM2.5. En el eje "x" se encuentra el tiempo mientras que en el eje "y" la concentración del contaminante. En la parte superior derecha está la leyenda, comprendiendo los valores del contaminante, el promedio móvil y el umbral, pudiendo cambiar de manera dinámica entre ellas.
Código de creación	<code>SELECT last("avg_pm25") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none)</code> <code>SELECT moving_average(mean("avg_pm25"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</code>

Tabla 6.24. Panel, gráfico valores PM25. Fuente: Elaboración propia

Valores de PM10 (Partículas en suspensión)

Se muestra la gráfica perteneciente al apartado referido al material particulado con diámetro igual o menor a 10 micrómetros. La estructura de paneles es similar a las anteriores.

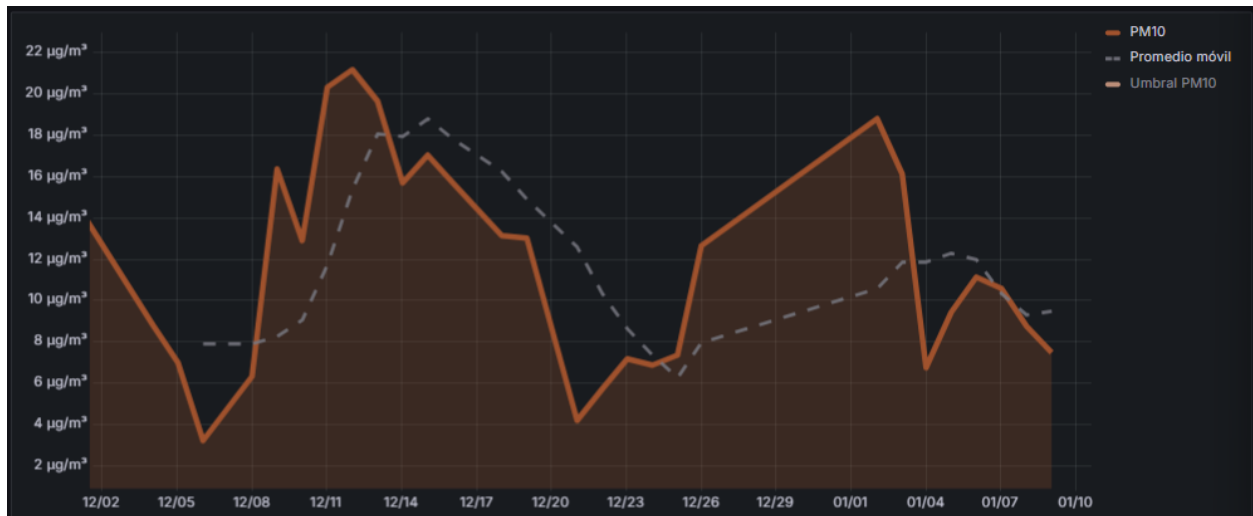


Figura 6.8. Grafana, apartado PM10. Fuente: Elaboración propia

PCA21	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor máximo de partículas en suspensión cuyo diámetro es menor a 10 micrómetros en el lugar y tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_pm10") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.25. Panel, valor máximo PM10. Fuente: Elaboración propia

PCA22	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor mínimo registrado de PM10 en el lugar y tiempo correspondientes. Aparece en color azul.
Código de creación	<code>SELECT min("avg_pm10") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.26. Panel, valor mínimo PM10. Fuente: Elaboración propia

PCA23	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Representa el valor promedio de todos los valores almacenados en el lugar y tiempo señalados de las concentraciones de partículas PM10 registradas. Aparece en color verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_pm10") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.27. Panel, valor promedio PM10. Fuente: Elaboración propia

PCA24	GRÁFICO VALORES PM10
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Gráfico dedicado a PM10. En el eje “x” se encuentra el tiempo mientras que en el eje “y” la concentración. En la parte superior derecha está la leyenda, comprendiendo las mediciones del contaminante, el promedio móvil y el umbral de PM10, pudiendo cambiar de manera dinámica entre ellas.
Código de creación	<pre>SELECT last("avg_pm10") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none) SELECT moving_average(mean("avg_pm10"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</pre>

Tabla 6.28. Panel, gráfico valores PM10. Fuente: Elaboración propia

Valores de O3 (Ozono)

A continuación, se exploran las visualizaciones correspondientes al elemento contaminante O3 (Ozono), recordando que en la capa más cercana a la tierra este gas no es beneficioso, considerándose contaminante. Se observan los paneles dedicados y su análisis de este apartado.

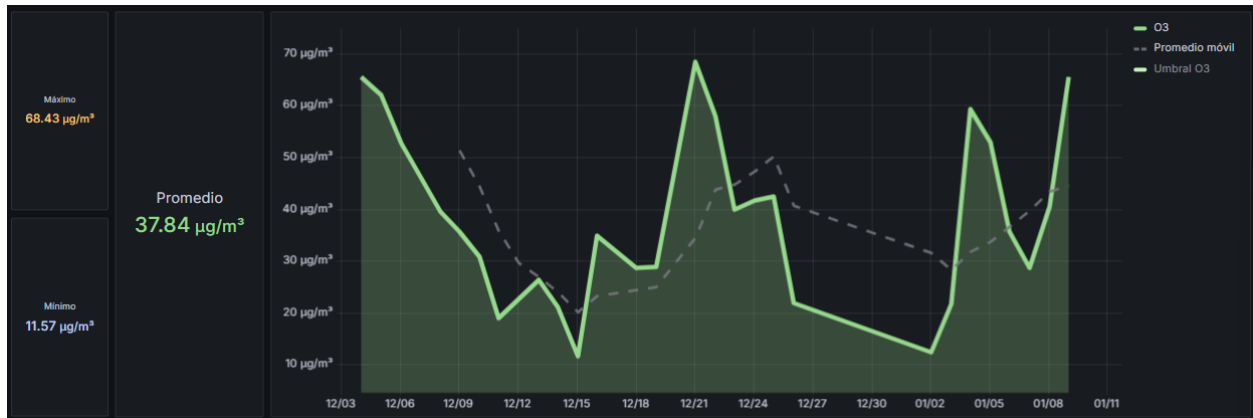


Figura 6.9. Grafana, apartado O3. Fuente: Elaboración propia

PCA25	VALOR MÁXIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor máximo de ozono troposférico en el lugar y tiempo seleccionados. Se muestra en color naranja.
Código de creación	<code>SELECT max("avg_o3") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.29. Panel, valor máximo O3. Fuente: Elaboración propia

PCA26	VALOR MÍNIMO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Muestra el valor mínimo registrado de O3 en el lugar y tiempo correspondientes. Aparece en azul.
Código de creación	<code>SELECT min("avg_o3") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.30. Panel, valor mínimo O3. Fuente: Elaboración propia

PCA27	VALOR PROMEDIO
Tipo de panel	Stat
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Representa el valor medio de ozono. Aparece en verde y a la derecha de los dos paneles anteriores.
Código de creación	<code>SELECT mean("avg_o3") FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter</code>

Tabla 6.31. Panel, valor promedio. Fuente: Elaboración propia

PCA28	GRÁFICO VALORES O3
Tipo de panel	Time Series
Unidad de medida	Microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$)
Descripción	Panel con la gráfica. En el eje “x” se encuentra el tiempo mientras que en el eje “y” la concentración. La leyenda comprende el contaminante, el promedio móvil y el umbral de O3.
Código de creación	<pre>SELECT last("avg_o3") FROM "aire" WHERE ("provincia"::tag =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(none) SELECT moving_average(mean("avg_o3"), 5) FROM "aire" WHERE ("provincia" =~ /^\$provincia\$/) AND \$timeFilter GROUP BY time(1d) fill(null)</pre>

Tabla 6.32. Panel, gráfico valores O3. Fuente: Elaboración propia

6.2.2.3 Visualización combinada

Esta visualización representa de forma conjunta los datos de calidad del aire y las variables meteorológicas, con el objetivo de facilitar su análisis integrado en un único elemento visual.

La información se muestra mediante una tabla estructurada por fechas, que actúan como eje principal de la visualización. Para cada fecha se incluyen, en columnas independientes los valores correspondientes a los distintos contaminantes atmosféricos monitorizados. Las dos últimas columnas están dedicadas a la temperatura máxima y mínima, lo que permite observar su comportamiento junto a los niveles de contaminación registrados. En cada celda de estas dos últimas columnas descritas anteriormente, se incluye un gráfico con una barra indicativa, representando las temperaturas máximas y mínimas.

Este formato tabular facilita la comparación directa entre contaminantes en un mismo instante temporal y permite analizar de forma simultánea la posible relación entre las condiciones

térmicas y la calidad del aire. Además, la disposición en filas cronológicas contribuye a identificar patrones, variaciones y episodios relevantes de manera clara y ordenada. Al pie de tabla tenemos una fila adicional que muestra los valores máximos por columna para un rápido acceso visual.

En la primera columna aparecen las fechas en formato AAAA/MM/DD, pensado para poder ser ordenado cronológicamente con facilidad al coincidir el orden textual con el temporal, ya que organiza de la unidad mayor (año) a la menor (día). Esto evita conversiones adicionales y facilita el tratamiento automático de fechas. Gracias a ello la tabla representada tiene la capacidad de ser ordenada ascendente o descendientemente según el criterio del usuario.

Se muestra una captura del dashboard con el apartado Matriz de contaminantes - temperatura que contiene la tabla combinada.

> Temperaturas (5 panels)

> Valores de NO2 (Dióxido de Nitrógeno) (4 panels)

> Valores de NO (Monóxido de Nitrógeno) (4 panels)

> Valores de SO2 (Dióxido de azufre) (4 panels)

> Valores CO (Monóxido de carbono) (4 panels)

> Valores de PM2.5 (Partículas en suspensión) (4 panels)

> Valores de PM10 (Partículas en suspensión) (4 panels)

> Valores de O3 (Ozono) (4 panels)

< Matriz de contaminantes - temperatura

Fecha	NO ₂ (µg/m³)	NO(µg/m³)	SO ₂ (µg/m³)	CO(mg/m³)	PM2.5(µg/m³)	PM10(µg/m³)	O ₃ (µg/m³)	Temp Max	Temp Min
2025/12/08	15.1	4.09	3.50	0.350	5	6.33	39.6	13.6	5.21
2025/12/09	22.1	8.10	3.50	0.400	8.75	16.3	35.7	13.7	5.18
2025/12/10	22.8	7.36	3	0.450	6.80	12.9	30.9	10.2	6.88
2025/12/11	25.1	13.4	3.50	0.400	12	20.3	19	11.6	4.92
2025/12/12	27	12.8	4	0.300	11.5	21.2	22.6	10.6	5.86
2025/12/13	20.7	8.18	4	0.350	12.8	19.7	26.3	13.1	2.77
2025/12/14	19.4	6.18	3.50	0.300	11.7	15.7	21	9.63	-0.270
2025/12/15	22.1	16.9	4.50	0.350	11	17	11.6	-	-
2025/12/16	17.2	6.18	4.50	0.300	6	15.7	34.8	9.13	4.30
2025/12/18	18.4	12.9	1.50	0.300	8.40	13.1	28.7	-	-
-----	---	---	-	---	---	---	---	---	---
Max	27	31.4	8.50	0.450	12.8	21.2	68.4		

Figura 6.10. Grafana, apartado matriz de contaminantes-temperatura. Fuente: Elaboración propia

Mostramos el análisis de este panel, teniendo en cuenta que aparecerán todas las mediciones y datos de los siguientes apartados en una misma matriz. El ID corresponderá a panel de calidad y temperatura (PCTx).

6.2.3. Interacción y navegación

En este apartado se describen los mecanismos de interacción incorporados en el dashboard, orientados a facilitar la exploración dinámica de los datos y a adaptar las visualizaciones a las necesidades del usuario.

La navegación se apoya en el uso de filtros globales que permiten seleccionar de forma dinámica la provincia de Castilla y León y el rango temporal de interés. Estos parámetros actúan de manera conjunta sobre todos los paneles del dashboard , de modo que cualquier modificación se refleja automáticamente en el conjunto de las visualizaciones. En las capturas se muestran ambos filtros de selección desde el dashboard.

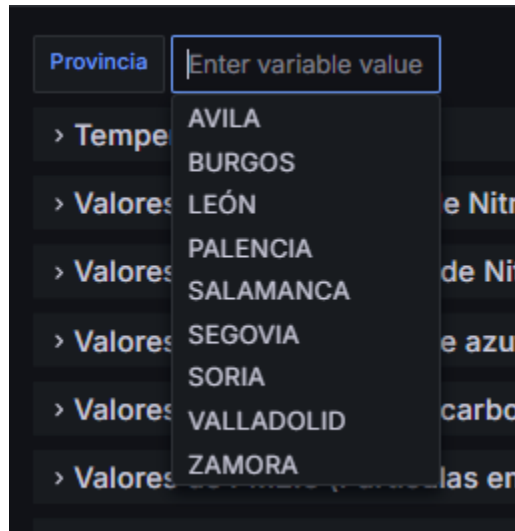


Figura 6.11. Filtro de provincias del dashboard Grafana. Fuente: Elaboración propia

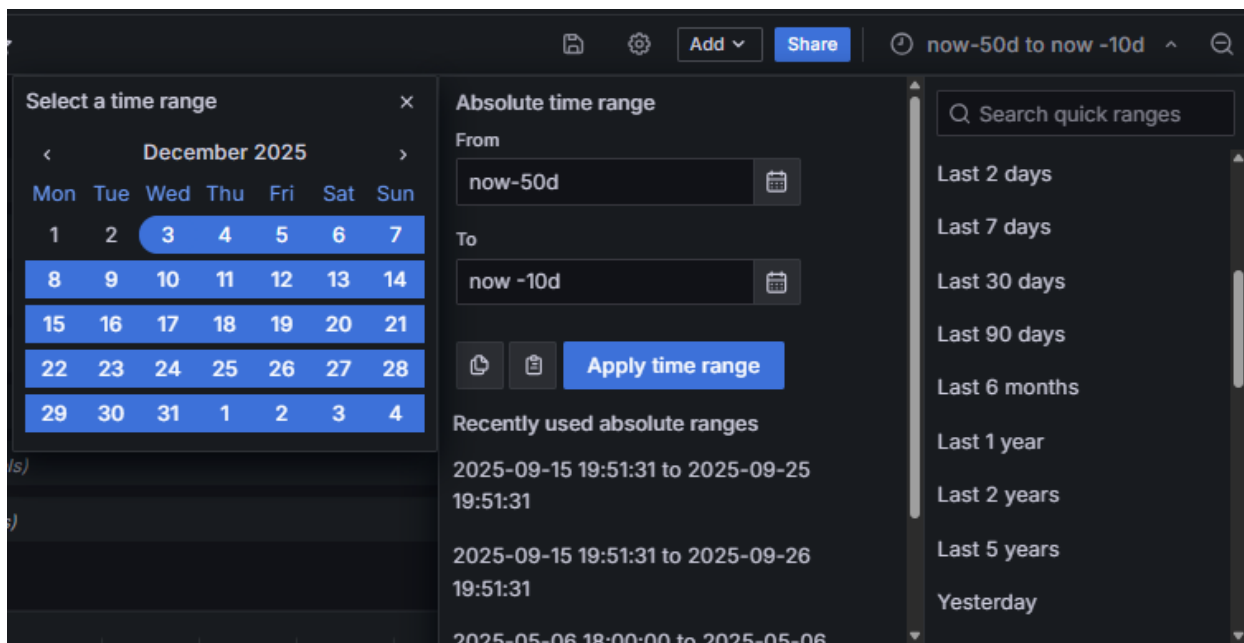


Figura 6.12. Filtro para rango de tiempo en dashboard Grafana. Fuente: Elaboración propia

Aparte de las variables a partir de las cuales se generan las visualizaciones dinámicamente y el usuario utiliza como filtros, existen otros elementos interactivos. En los paneles de tipo serie temporal dedicados a los contaminantes, la leyenda interactiva permite gestionar la visualización de las distintas variables representadas. El usuario puede aislar una variable concreta o visualizar varias de forma simultánea, lo que facilita el análisis comparativo y la interpretación de su evolución temporal. A continuación, observamos el panel de serie temporal del ozono en el que se “juega” con la leyenda activando las diferentes visualizaciones de los valores independientemente. Con la tecla “ctrl” del teclado pulsada se hace click en los elementos de la leyenda y se visualizan u ocultan, cómo se puede observar en las imágenes siguientes.

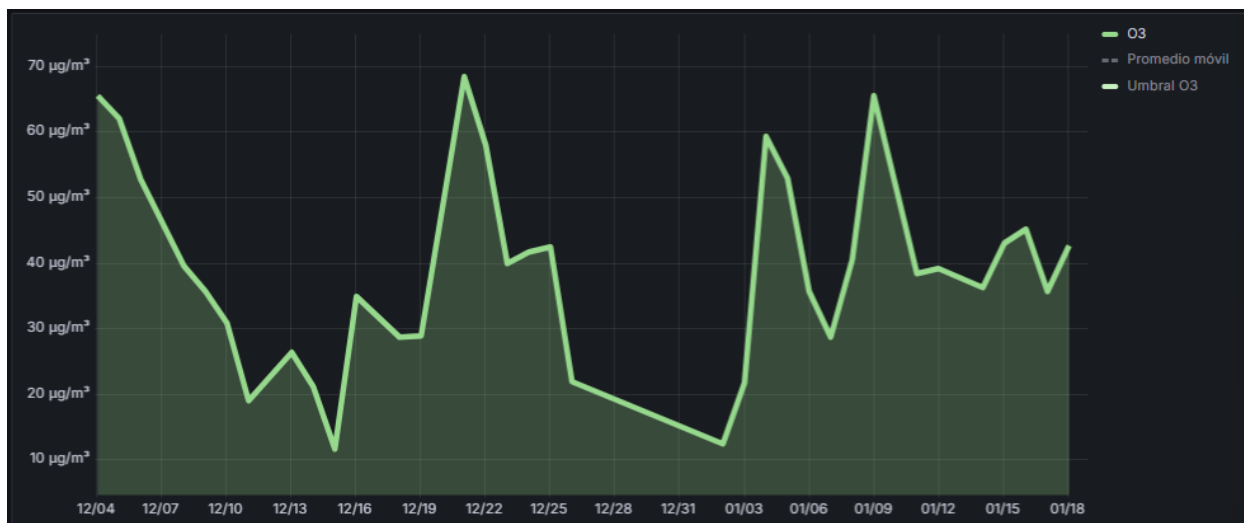


Figura 6.13. Grafana, solo valores O₃ visibles. Fuente: Elaboración propia

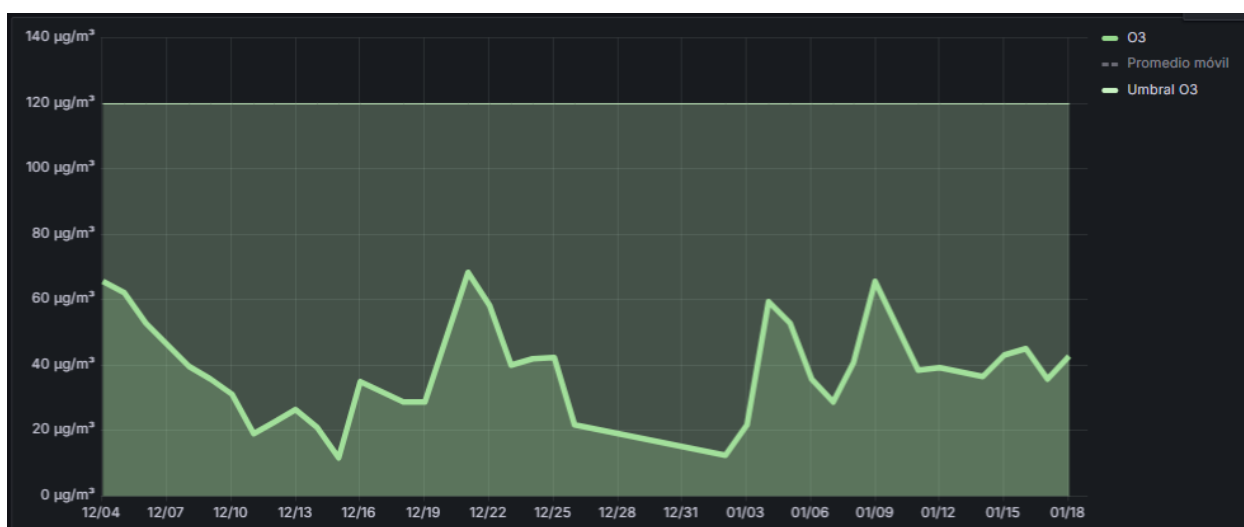


Figura 6.14. Grafana, valores O₃ + valor umbral visibles. Fuente: Elaboración propia

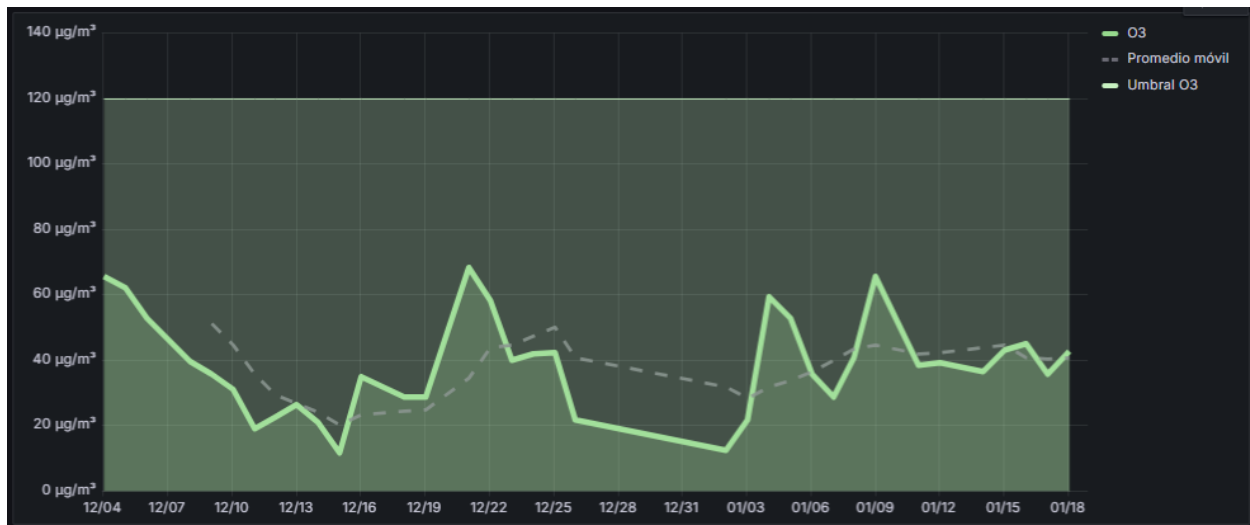


Figura 6.15. Grafana, valores O3 + umbral + promedio móvil. Fuente: Elaboración propia

En cuanto al último apartado desplegable, matriz de contaminantes - temperatura el usuario podrá interactuar de manera dinámica ordenando ascendente o descendientemente los valores de cada columna, incluyendo la fecha situada en la primera. La tabla también posee una barra de scroll (o barra de desplazamiento), para poder ver el contenido que excede los límites de la tabla y del dashboard.

Capítulo 7

7. Conclusiones y trabajos futuros

En este trabajo se ha desarrollado un sistema completo para la obtención, procesamiento, almacenamiento y visualización de los datos ambientales, centrado en el análisis de calidad del aire y variables meteorológicas en la comunidad de Castilla y León. El sistema integra distintas tecnologías actuales, organizadas y desplegadas mediante una arquitectura basada en contenedores independientes, lo que ha permitido una solución modular y con alto nivel de escalabilidad.

Los objetivos planteados al inicio del trabajo se han cumplido satisfactoriamente. Se ha conseguido automatizar la adquisición de datos desde diferentes fuentes oficiales, aplicar procesos de transformación adecuados, y un sistema tipo Datalake para el almacenamiento de la información deseada sin grandes restricciones. Asimismo, la fase de exploración ha permitido presentar los datos de manera clara e intuitiva, facilitando su interpretación a través de visualizaciones dinámicas.

La solución desarrollada demuestra la viabilidad del uso de herramientas de código abierto para la construcción de sistemas de monitorización ambiental. La combinación de un almacenamiento específico orientado a series temporales y paneles interactivos, permite analizar la evolución de los contaminantes y variables térmicas de forma flexible, aportando valor tanto desde un punto de vista técnico como analítico.

En conjunto, el sistema presentado constituye una base sólida para el desarrollo y despliegue de soluciones de gestión de datos ambientales, apoyándose en una arquitectura completamente dockerizada que facilita su instalación, mantenimiento y portabilidad. Este enfoque permite reproducir el entorno de manera sencilla, garantizar la coherencia entre ejecuciones y simplificar tanto el despliegue como la escalabilidad del sistema, proporcionando una plataforma robusta y fácilmente adaptable a futuros escenarios o ampliaciones.

7.1 Trabajos futuros

A partir del sistema desarrollado, se abren diversas líneas de trabajo que permiten ampliar y mejorar la solución presentada. Una primera línea clara consiste en la incorporación de nuestras variables ambientales, tanto meteorológicas como de calidad del aire, lo que permitiría enriquecer el análisis y ofrecer una visión más completa del entorno monitorizado.

La arquitectura basada en contenedores facilita la integración de nuevas tecnologías sin afectar al funcionamiento del sistema existente. Esta característica permite, por ejemplo, añadir nuevos servicios de almacenamiento, procesamiento o análisis de datos de forma sencilla, aprovechando la modularidad y flexibilidad que ofrece la dockerización.

Otra posible ampliación del sistema sería la extensión del ámbito geográfico, incorporando datos de más provincias o incluso de otras comunidades autónomas. Esta ampliación permitiría realizar análisis a mayor escala y aumentar el valor del sistema como herramienta de monitorización ambiental.

Adicionalmente, podrían incorporarse mecanismos más avanzados, como la adición de técnicas de análisis predictivo o la generación de alertas ante determinados eventos, aplicables a los elementos contaminantes de calidad del aire. Estas mejoras podrían contribuir a transformar el sistema y orientarlo más a la toma de decisiones.

Bibliografía

[1] BBC Mundo. El mapa que muestra cuán contaminado está el aire que respiramos. Disponible en <https://www.bbc.com/mundo/noticias-37501943>. Visitado por última vez el 20 de noviembre de 2026.

[2] Docker. Docker: Accelerated Container Application Development. Disponible en <https://www.docker.com/>. Visitado por última vez el 25 de noviembre de 2026.

[3] HolaMundo. Aprende Docker ahora! curso completo gratis desde cero!, YouTube, 7 jul. 2022. Disponible en <https://www.youtube.com/watch?v=4Dko5W96WHg>. Visitado por última vez el 29 de enero de 2026.

[4] The Apache Software Foundation. Apache NiFi. Disponible en <https://nifi.apache.org/>. Visitado por última vez el 29 de enero de 2026.

[5] Apache Nifi - 1. Inicio rápido, YouTube, disponible en <https://www.youtube.com/watch?v=pmqTB1ytYuQ>. Visitado por última vez el 29 de enero de 2026.

[6] Oscar Fernandez. Apache NiFi: Introducción, Aprender Big Data, 08 mar. 2025. Disponible en <https://aprenderbigdata.com/introduccion-apache-nifi/>. Visitado por última vez el 29 de enero de 2026.

[7] MinIO. MinIO: High Performance Object Storage. Disponible en <https://www.min.io/>. Visitado por última vez el 29 de enero de 2026.

[8] NullSafeArchitect. Tutorial MinIO: Domina la solución para persistencia de datos, YouTube, disponible en <https://www.youtube.com/watch?v=XQrX-jQk5zM>. Visitado por última vez el 29 de enero de 2026.

- [9] DataCamp. MinIO Docker: Setup Guide for S3-Compatible Object Storage. Disponible en <https://www.datacamp.com/tutorial/minio-docker>. Visitado por última vez el 29 de enero de 2026. :contentReference[oaicite:0]{index=0}
- [10] InfluxData. InfluxDB: Time Series Platform for Metrics and Events. Disponible en <https://www.influxdata.com/>. Visitado por última vez el 29 de enero de 2026.
- [11] OpenWebinars. ¿Qué es InfluxDB y primeros pasos?, OpenWebinars Blog. Disponible en <https://openwebinars.net/blog/que-es-influxdb-y-primeros-pasos/>. Visitado por última vez el 29 de enero de 2026.
- [12] InfluxData. InfluxDB Line Protocol Tutorial. Disponible en https://docs.influxdata.com/influxdb/v1/write_protocols/line_protocol_tutorial/. Visitado por última vez el 29 de enero de 2026.
- [13] Grafana Labs. Grafana: Open Source Analytics & Monitoring. Disponible en <https://grafana.com/>. Visitado por última vez el 29 de enero de 2026.
- [14] InfluxData. Getting started with InfluxDB and Grafana. Disponible en <https://www.influxdata.com/blog/getting-started-influxdb-grafana/>. Visitado por última vez el 29 de enero de 2026.
- [15] Wikipedia. Monóxido de carbono, Wikipedia, disponible en https://es.wikipedia.org/wiki/Mon%C3%B3xido_de_carbono. Visitado por última vez el 29 de enero de 2026.
- [16] Ministerio para la Transición Ecológica y el Reto Demográfico. Óxidos de nitrógeno, glosario de términos. Disponible en <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/glosario-de-terminos/glosario-contaminantes/oxidos-nitrogeno.html>. Visitado por última vez el 29 de enero de 2026.

[17] NASA Scientific Visualization Studio. NASA images show human fingerprint on global air quality. Disponible en <https://svs.gsfc.nasa.gov/12094>. Visitado por última vez el 29 de enero de 2026.

[18] Èlia Pons. El ozono, un contaminante del aire con dos caras. Disponible en <https://www.isglobal.org/healthisglobal/-/custom-blog-portlet/el-ozono-un-contaminante-del-aire-con-dos-caras>. Visitado por última vez el 29 de enero de 2026.

[19] Ministerio para la Transición Ecológica y el Reto Demográfico. Dióxido de azufre, glosario de términos. Disponible en <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/glosario-de-terminos/glosario-contaminantes/dioxido-azufre.html>. Visitado por última vez el 29 de enero de 2026.

[20] Servicio Murciano de Salud. Materia particulada. Disponible en <https://www.murciasalud.es/web/sanidad-ambiental/materia-particulada>. Visitado por última vez el 29 de enero de 2026.

[21] Wikipedia Commons. Mapa global de PM2.5. Disponible en https://upload.wikimedia.org/wikipedia/commons/4/4f/483897main_Global-PM2.5-map.JPG. Visitado por última vez el 29 de enero de 2026.

[22] IQAir. Wind, weather and air pollution. Disponible en <https://www.iqair.com/es/newsroom/wind-weather-air-pollution>. Visitado por última vez el 29 de enero de 2026.

[23] Ministerio para la Transición Ecológica y el Reto Demográfico. Índice de Calidad del Aire (ICA). Disponible en <https://ica.miteco.es/>. Visitado por última vez el 29 de enero de 2026.

[24] IronSoftware. Docker Compose network, IronSecuredOC Blog. Disponible en <https://ironsoftware.com/es/enterprise/securedoc/blog/using-ironsecuredoc/docker-compose-network/>. Visitado por última vez el 29 de enero de 2026.

[25] Junta de Castilla y León. Carga de datos históricos de calidad del aire. Disponible en <https://servicios.jcyl.es/esco/cargarFrmDatosHistoricos.action>. Visitado por última vez el 29 de enero de 2026.

[26] Agencia Estatal de Meteorología (AEMET). AEMET OpenData: API de datos meteorológicos. Disponible en <https://opendata.aemet.es/>. Visitado por última vez el 29 de enero de 2026.

[27] Organización Mundial de la Salud (OMS). Disponible en <https://www.who.int/es>. Visitado por última vez el 29 de enero de 2026.

[28] Unión Europea. Directiva 2008/50/CE, de 21 de mayo de 2008, relativa a la calidad del aire ambiente y a una atmósfera más limpia en Europa. DOUE L 152 de 11 de junio de 2008. Disponible en <https://www.boe.es/buscar/doc.php?id=DOUE-L-2008-81053>. Visitado por última vez el 29 de enero de 2026.

[29] Ministerio de la Presidencia. Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire, Enero 2011. Publicado en BOE núm. 25, de 29 de enero de 2011, páginas 9574 a 9626. Disponible en <https://www.boe.es/buscar/act.php?id=BOE-A-2011-1645>. Visitado por última vez el 29 de enero de 2026.

[30] Código Electrónica. Hablemos Docker. Disponible en <http://codigoelectronica.com/blog/hablemos-docker>. Visitado por última vez el 29 de enero de 2026.