Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo Fin de Máster

Máster en Ingeniería Informática - Especialidad en Big Data

Control, análisis y predicción de la calidad del aire en Valladolid mediante tecnología Big Data

Autora:
Esther Cuervo Fernández

Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo Fin de Máster

Máster en Ingeniería Informática - Especialidad en Big Data

Control, análisis y predicción de la calidad del aire en Valladolid mediante tecnología Big Data

Autora:
Esther Cuervo Fernández

Tutores:
Miguel Ángel Martínez Prieto
Anibal Bregon Bregon

Resumen

La presencia de contaminantes en el aire se sitúa como una de las mayores causas de muerte prematura en el mundo, con el 91 % de la población humana viviendo en ambientes con niveles de polución peligrosos. Además, la reducción de los niveles de contaminante es uno de los Objetivos de Desarrollo Sostenible de las Naciones Unidas, por lo que es importante para gobiernos y entidades locales contar con un sistema complejo de control de la calidad del aire. Uno de los primeros pasos para ello es tener una amplia red de medición, como es el caso en la provincia de Valladolid, que además ofrece sus datos de manera abierta. Sin embargo, no dispone de sistemas de exploración de datos y predicción, con los que se mejoraría la transparencia y fomentaría la creación de estudios sobre el tema.

En este Trabajo Fin de Máster se plantea la creación de un sistema Big Data que permita el control, análisis y predicción de datos horarios de once estaciones de medición de la calidad del aire en la provincia de Valladolid, utilizando datos totalmente públicos, con una metodología iterativa apoyada en un proceso ETL completamente automatizado.

El resultado es un Data Lake con actualización diaria implementado sobre un clúster Hadoop real, una herramienta de visualización elaborada en PowerBI, que proporciona un *dashboard* con la información necesaria para realizar un seguimiento diario de la calidad del aire, y un análisis exploratorio y predictivo, utilizando algoritmos de Aprendizaje Automático, con resultados satisfactorios respecto a la capacidad de producir predicciones certeras en múltiples puntos de la provincia a partir de los datos recogidos.

Abstract

The presence of pollutants in the air is currently one of the top causes for premature deaths in the world, with 91% of the world's population living with dangerous levels of air pollution. Besides this, the reduction of pollution levels is one of the Sustainable Development Goals of the United Nations, which makes it key for governments and local authorities to have a complex system for air quality control. One of the first steps towards this goal is having a vast measurement network, as is the case in Valladolid, whose data is also offered openly. However, there's a lack of data exploration systems and predictions in the resources that Valladolid offers, with which transparency could be improved, besides also sponsoring the creation of new projects around the theme.

In this End of Master's Project we set up the creation of a Big Data system that would allow for the control, analysis and prediction of hourly pollution data from eleven measurement stations around the province of Valladolid, using data that is completely public, with an iterative methodology based on am entirely automated ETL process.

The result is a Data Lake updated daily deployed over a real Hadoop cluster, a visualization tool implemented in PowerBI, providing a dashboard with the information needed to follow up on daily air quality, and lastly an exploratory and predictive analysis using Machine Learning algorithms, with satisfactory results with regards to the capacity of producing accurate predictions in multiple points of the province from the data used.

Índice general

Índice de figuras

Índice de cuadros

Listados de código

Capítulo 1

Introducción

Este capítulo presenta el contexto que enmarca el proyecto, la motivación del mismo, los objetivos a alcanzar y la estructura del resto del documento.

1.1. Contexto

La problemática de la calidad del aire es, junto al cambio climático y fuertemente relacionado con este, una de las mayores preocupaciones a nivel internacional. La razón es simple: la exposición a estos contaminantes causó un estimado de 4,2 millones de muertes prematuras en 2016 [3] dados sus efectos agravantes de enfermedades coronarias y respiratorias, incluido cáncer de pulmón [4].

La Organización Mundial de la Salud (OMS) estableció en 2016 medidas para responder de manera global a la crisis de la polución del aire, sin embargo ese mismo año, el 91 % de la población vivía en lugares donde no se cumplían los máximos de polución establecidos por la OMS [3]. Como se puede ver el la Figura 1.1 el problema es especialmente crítico en ciertas zonas del mundo como Asia y algunos países de África.

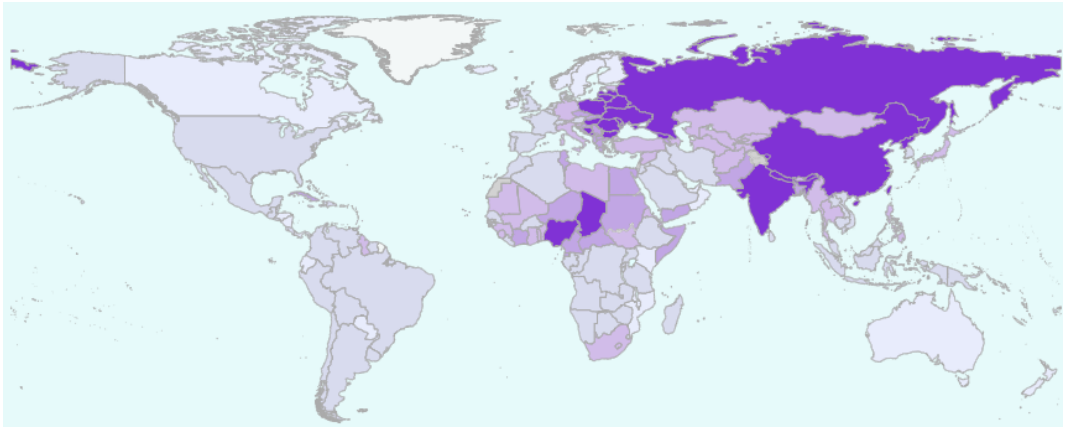


Figura 1.1: Número de muertes atribuibles a la polución por cada 100.000 habitantes en 2016 - Fuente: https://www.who.int/gho/phe/outdoor_air_pollution/burden/en/

La mayoría de orígenes de mala calidad del aire tienen que ver con lo que se conoce como energía «sucia», principalmente la quema de combustibles fósiles en transporte, industria, calefacción, agricultura, etc. por tanto está clara su conexión con el cambio climático, ya que estas fuentes son también aquellas que causan el mayor impacto en el calentamiento global y en la producción de gases de efecto invernadero.

Dentro de los objetivos y metas de desarrollo sostenible establecidos por las Naciones Unidas [5], la calidad del aire juega un papel en los objetivos 3 - *Salud y bienestar*, 11 - *Ciudades y comunidades sostenibles* y 13 - *Acción por el clima*, así como el objetivo 7.2 - *Acceso de los hogares a energía limpia*. La calidad del aire tiene su propio objetivo, 11.6 - *Reducción del impacto de las ciudades en el entorno*, con especial atención a la calidad del aire [6]. El indicador medido para su consecución es el PM2,5 (partículas con menos de 2,5 micrómetros de diámetro), ya que es capaz de penetrar profundamente en el cuerpo humano, causando los mayores daños.

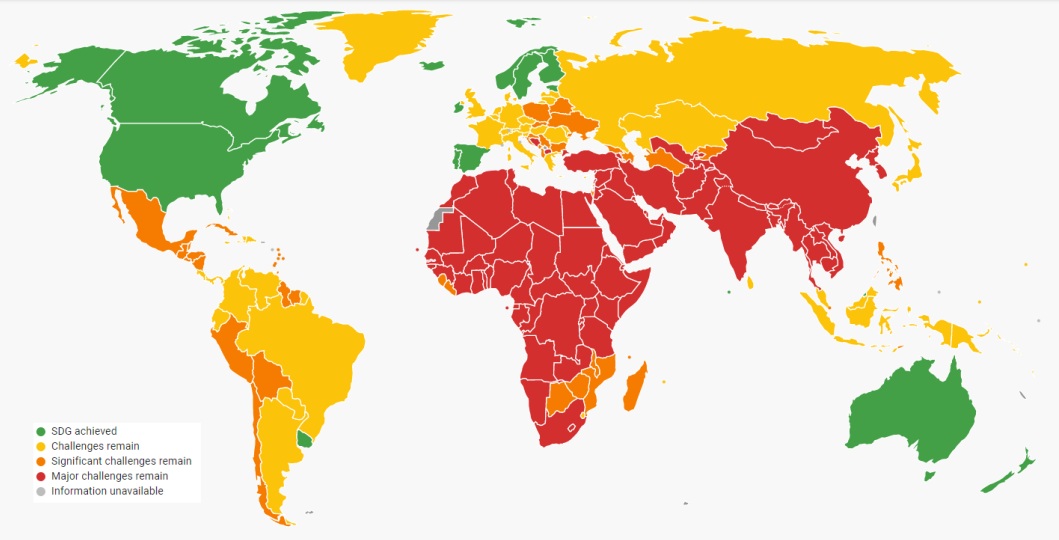


Figura 1.2: Consecución por países del objetivo 11.6 - Fuente: https://dashboards.sdgindex.org/map/indicators/sdg11_pm25

Como se puede observar, hay muy pocos países que hayan conseguido alcanzar este objetivo completamente, aunque España se encuentra entre los que han reducido las cantidades de PM2,5 por debajo del umbral establecido.

1.2. Motivación

Como se ha visto en la Figura 1.2, España ha conseguido alcanzar la meta de reducir el PM2,5 por debajo de el límite marcado por la Organización Mundial de la Salud, sin embargo, el informe de evaluación de la calidad del aire en España de 2019 aún señala preocupantes superaciones de los umbrales establecidos por directivas Europeas [1]. La gran problemática de la calidad del aire en España es la presencia de Ozono troposférico, que también se relaciona con mayor mortalidad.

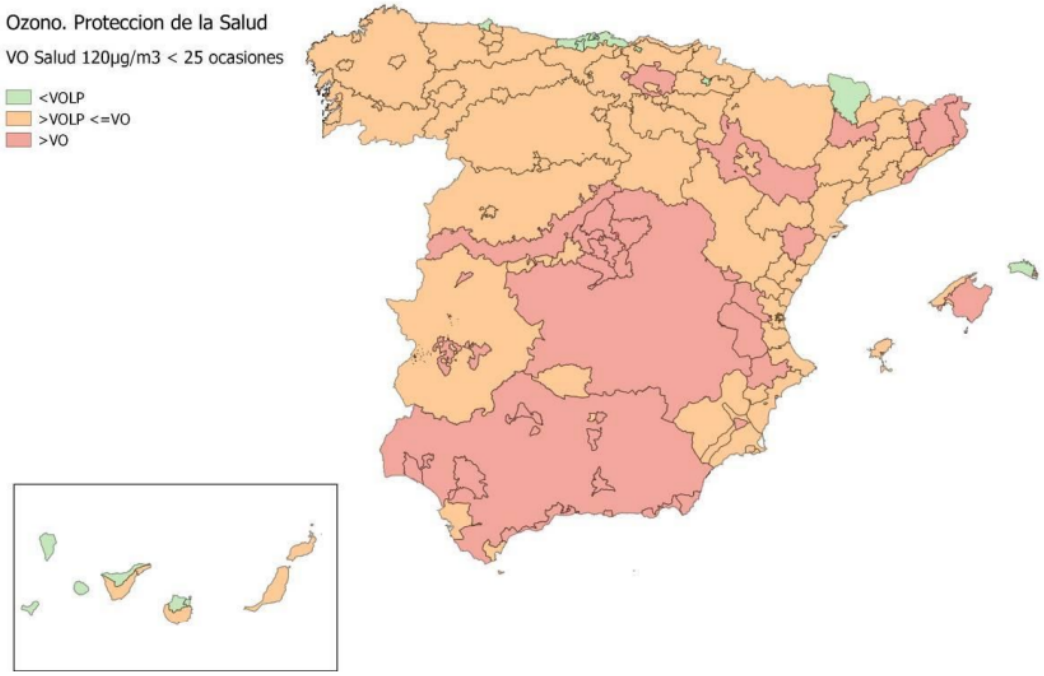


Figura 1.3: Superación de valores objetivos a largo plazo (VOLP) y corto plazo (VO) en España - Fuente: Informe de evaluación de la calidad del aire [1]

Poniendo el foco en la provincia de estudio de este trabajo: Valladolid, esta ha implantado en los últimos años medidas urgentes por mala calidad del aire en la ciudad, entre ellas restricciones de movilidad de vehículo . Sin embargo, en especial en 2019, la problemática de la contaminación en Valladolid se agravó [7], con expertos neumólogos advirtiendo del peligro de los datos de contaminación para la salud de los ciudadanos y la saturación de hospitales [8].

Aunque existen, como veremos más adelante, múltiples herramientas para el control de la calidad del aire a nivel nacional e internacional, el ayuntamiento de Valladolid aún tiene serias deficiencias para ofrecer un sistema de visualización y exploración de sus datos eficaz, así como una falta de predicciones disponibles, material útil para colectivos de riesgo que deseen conocer los valores esperados de contaminación en Valladolid para su propia protección, o para investigadores interesados en realizar estudios de la calidad del aire local.

Sin embargo, de manera positiva, la red de control de la contaminación de Valladolid ofrece un número alto de estaciones de medida, así como un acceso a los datos bastante homogéneo a través de portales de la Junta de Castilla y León.

Por esto, se ha decidido utilizar los datos de esta ciudad como fuente de estudio, aplicando tecnología Big Data debido al alto volumen de datos históricos disponible, así como el rápido aumento de los datos, sobre todo si se considera una posible ampliación de este trabajo para considerar todo Castilla y León.

1.3. Objetivos

El objetivo principal de este trabajo es utilizar los datos de calidad del aire disponibles para la ciudad de Valladolid, así como otras fuentes de datos valiosas, para crear visualizaciones útiles para expertos y conocedores de la problemática de la calidad del aire, así como elaborar un análisis y predicción de los valores horarios esperados de contaminantes en cada punto de medición en la provincia.

Para ello se hará uso de una estructura Data Lake, proceso ETL y aprendizaje automático, surgiendo múltiples subobjetivos que podemos evaluar durante la realización del trabajo:

1. Creación de un Data Lake para la ingesta, almacenamiento, transformación y carga de los datos, destinado a disponer de un repositorio central de información utilizable para los siguientes dos objetivos.
2. Elaborar una visualización que permita conocer los valores de calidad del aire en Valladolid en las últimas 24 horas, así como otras variables que puedan estar relacionadas, como la situación meteorológica.
3. Realizar un análisis exploratorio de los datos que nos permita conocer qué variables son las más útiles para la predicción de la calidad del aire.
4. Crear modelos de aprendizaje automático que permitan predecir los datos horarios en cada estación de recogida a partir del histórico, y evaluar la precisión de dichos modelos según las métricas tradicionales.

La consecución de estos objetivos se evaluará en el capítulo 8.

1.4. Estructura de capítulos

El documento se divide en los siguientes capítulos:

- **Capítulo 2: Marco teórico y estado del arte** en el que se sientan las bases teóricas necesarias para entender la problemática del control de la calidad del aire, así como la exploración de las herramientas Big Data existentes a nivel internacional para el control de la calidad del aire, y por último, el fundamento de las técnicas de aprendizaje automático utilizadas en la etapa de predicción.
- **Capítulo 3: Metodología y arquitectura de la solución** donde se detalla la metodología de trabajo usada, y un repaso a las herramientas y arquitectura que conforma tanto el Data Lake como el entorno en el que se realiza la explotación de los datos.
- **Capítulo 4: Análisis y Diseño** en el cual se determina los modelos que definen los datos refinados, las fuentes de datos de las que disponemos, y las transformaciones necesarias para llegar a los datos refinados a partir de los datos en bruto.
- **Capítulo 5: ETL** se trata de una descripción de la implementación del proceso ETL, cuyo modelado se lleva a cabo en el capítulo anterior.

- **Capítulo 6: Exploración** detalla el proceso de planteamiento e implementación de la visualización para el control de la calidad del aire
- **Capítulo 7: Explotación** se llevan a cabo los análisis exploratorio y predictivo, así como la evaluación de los modelos resultantes.
- **Capítulo 8: Conclusiones y trabajo futuro** en el que se presentan las conclusiones obtenidas del trabajo, y se enumeran algunas de las posibles lineas de trabajo futuro a partir de este proyecto.

Además se dispone de dos apéndices, ambos conteniendo el código fuente para la creación de tablas, en el primer apéndice en Hive y en el segundo en SQL Server.

Capítulo 2

Marco teórico y estado del arte

En este capítulo explicamos aquellos aspectos teóricos sobre el control de la calidad del aire que consideramos necesario conocer para la completitud de este trabajo, así como una exploración de algunas de las herramientas existentes a nivel internacional usando tecnología Big Data para crear visualizaciones, informes y predicciones relacionadas con la calidad del aire. Terminamos el capítulo asentando las bases teóricas de los algoritmos de aprendizaje automático utilizados en este trabajo para la predicción de contaminantes.

2.1. Control de la calidad del aire

El control de la calidad del aire es el mecanismo por el cual se realiza la medición, cálculo, predicción o estimación de aquellas sustancias presentes en el aire o en superficies que, bien por si solas o por reacciones químicas, son perjudiciales para la salud y el medio ambiente.

En esta sección se explora una descripción de los contaminantes utilizados en este trabajo, su origen y las consecuencias que pueden tener en la población y el medio ambiente, el efecto que tiene la meteorología de un lugar en los contaminantes que aparecen o se acumulan en dicha zona, y por último una descripción de la red de evaluación de la calidad del aire existente en Valladolid.

Las fuentes principales utilizadas en esta sección son: *Air Pollution* de D.H.F. Liu [9], la página web del Ministerio Para la Transición Ecológica y el Reto Demográfico del Gobierno de España [10, 11], *Air Pollution and heatlh* de Bert Brunekreef [12] y *Contaminación por monóxido de carbono: un problema de salud ambiental* de Jairo Téllez [13].

2.1.1. Contaminantes

Aunque existen múltiples sustancias dañinas, normalmente se miden sólo ciertos contaminantes, aquellos que son de especial preocupación por su grave efecto en la salud de la población y en el medio ambiente. Específicamente

para este trabajo se han tratado siete contaminantes, que se pueden dividir en: Ozono troposférico (O3), partículas en suspensión (PM10 y PM2,5) y gases (CO, NO, NO2 y SO2). Estos dos primeros son especialmente dañinos para la salud humana, mientras que los gases ocasionan principalmente daños al medio ambiente y cultivos.

Ozono troposférico El ozono que existe por debajo de la estratosfera (y ajeno a la famosa «capa de ozono») se encuentra en contacto con la superficie terrestre y es fuente de múltiples daños a la salud humana, ecosistemas y cultivos. Este contaminante no se emite directamente, si no que se forma por procesos que principalmente involucran a óxidos de nitrógeno, monóxido de carbono y metano, que reaccionan en presencia de la luz solar.

Hay múltiples reacciones químicas que pueden dar origen al ozono, aunque la más fácilmente detectable con los datos utilizados en este trabajo es la que involucra óxidos de nitrógeno:

1. El Monóxido de nitrógeno (NO) se oxida por el Dioxígeno (O2) produciendo Dióxido de nitrógeno (NO2). NO + O2 → NO2
2. En presencia de luz solar, el NO2 se disuelve de nuevo en NO y una sola partícula de Oxígeno (O). NO2 + luz →NO + O
3. Esta partícula de Oxígeno se une al Dioxígeno de la atmósfera para formar Ozono (O3). O + O2 →O3

Midiendo NO y NO2 junto a una variable que represente la insolación de una zona, podemos prever los niveles de O3 esperados.

Los efectos del O3 en la salud humana se centran en irritación e inflamación pulmonar, sobre todo durante ejercicio físico. También provoca daños en vegetación y cultivos que pueden ocasionar pérdidas significativas en agricultura [14].

Material particulado Este contaminante engloba una gran variedad de compuestos, los cuales pueden tener diferentes orígenes y efectos, sin embargo, en el caso de Europa, el control de las partículas en el aire se lleva a cabo según su tamaño, ya que es lo que determina si dicho material puede penetrar en las vías respiratorias.

Se consideran por tanto dos tipos de partículas: aquellas cuyo diámetro sea menor de 10 μm, siendo capaces de abrirse paso hacia las vías respiratorias, y englobadas en las anteriores las de diámetro menor de 2,5μm que pueden acceder a los alvéolos, parte del sistema respiratorio encargada del intercambio de oxígeno con la sangre. El tamaño de las partículas juega a su vez un papel en la facilidad de este contaminante de dispersarse o mantenerse suspendido, según si es de menor o mayor diámetro respectivamente.

Debido a su diversidad, el origen de este contaminante puede ser tanto natural (incendios forestales y erupciones volcánicas) como consecuencia de actividades humanas (actividad agrícola, de construcción o industrial). Sin embargo, las partículas de mayor tamaño suelen ser emitidas directamente, mientras que las de menor tamaño se producen por reacciones o procesos químicos a partir de gases.

Este tipo de contaminante parece causar efectos en la salud a cualquier nivel de exposición, aunque el riesgo aumenta con el contacto continuado con ellos. Por ello es de especial interés medir tanto la exposición breve

(concentración durante 24 horas) como prolongada (la media anual). Como ya se ha mencionado, suelen tener mayor afección a sistema respiratorio y cardiovascular.

Gases En el caso de los contaminantes medidos en el este trabajo, incluyen tanto gases acidificantes, aquellos que pueden comunicar propiedades ácidas, como gases eutrofizantes, los cuales causan contaminación en cuerpos de agua como ríos, lagos o embalses, provocando un exceso de nutrientes en el agua con resultados dañinos para el ecosistema acuático como la pérdida o aumento excesivo de seres vivos, pérdida de la calidad del agua, o aparición de toxinas causadas por ciertos tipos de algas. Fuera de ambos grupos se mide el Monóxido de carbono (CO).

Los gases acidificantes, entre ellos el Dióxido de azufre (SO2) y los Óxidos de nitrógeno (NO y NO2) regresan a la superficie por precipitaciones, causando daños en cultivos, oxidación acelerada de metales, y afecta negativamente a cualquier ecosistema sensible al ácido. Además son capaces de viajar grandes distancias por efecto del viento, así como permanecer en el aire durante varios días, pudiendo causar daños en zonas alejadas de su origen. Se producen por actividades industriales como combustión, y en el transporte.

Los gases eutrofizantes incluyen los Óxidos de nitrógeno y el metano, y sus fuentes son similares a las de los gases acidificantes.

Por último el Monóxido de carbono (CO) se genera en la combustión de combustibles con carbono (gas, petróleo, carbón, madera...) cuando no hay suficiente Oxígeno para formar CO2. Aparecen en combustión en sectores no industriales, y en cada vez menor medida en transporte por carretera. Puede provocar diversos daños en el sistema pulmonar, cardiovascular y nervioso tanto en humanos como en animales. También contribuye a la formación de gases de efecto invernadero, siendo uno de los gases precursores del Ozono .

2.1.2. Efecto de la meteorología

El estado meteorológico de una zona juega un papel importante en la creación, acumulación o dispersión de ciertos contaminantes, por lo que es importante conocer sus valores para tener una imagen completa de la calidad del aire esperada en una zona. Los indicadores más cruciales en esta tarea son: la velocidad y dirección del viento, la radiación solar, la estabilidad atmosférica, y las precipitaciones. A continuación hablamos brevemente sobre cada una y el efecto que pueden tener en los contaminantes:

Viento La distribución de la dirección del viento indica hacia qué áreas se transportan más frecuentemente los contaminantes, y la velocidad determina el tiempo que tarda en llegar un contaminante desde su fuente a un receptor, así como la cantidad de contaminante que se difunde en la dirección de dicho viento.

Radiación solar Como se ha visto anteriormente, la presencia de luz solar es crucial en los procesos que causan la formación del ozono troposférico, lo que provoca que este contaminante ocurra con mayor frecuencia durante el verano y primavera del hemisferio norte.

Estabilidad atmosférica Una atmósfera inestable es aquella donde los conjuntos de aire moviéndose hacia arriba no se enfrían lo suficientemente rápido, estando más calientes que el aire a su alrededor, causando que la masa de aire siga ascendiendo (ya que la densidad del aire disminuye según aumenta su temperatura). Durante este proceso la atmósfera es inestable, mientras que cuando la masa de aire se enfría más que su entorno y vuelve a descender, se trata de una atmósfera estable. Una atmósfera inestable es más propensa a las llamadas turbulencias atmosféricas, que son el mecanismo más eficaz para dispersar una nube de contaminación.

Otros efectos relacionados con la temperatura de la atmósfera son los conocidos como inversiones térmicas, partes de la atmósfera donde la temperatura aumenta con la altitud en lugar de disminuir. Las inversiones de importancia para la contaminación son las inversiones por radiación y por subsidencia:

- *Inversión por radiación:* Se produce a niveles bajos de altitud y se disipa rápidamente, ocurre durante periodos de clima despejado y vientos suaves. Es causada por un enfriamiento rápido de la tierra durante el anochecer, lo que causa que el aire en contacto con la tierra se enfríe más que el aire el alturas superiores. Este efecto continua hasta el día siguiente, cuando la tierra se vuelve a calentar por la acción solar. La polución emitida durante la noche queda atrapada en el aire durante esta inversión.
- *Inversión por subsidencia*: Afecta a grandes áreas durante múltiples días, reduciendo la capacidad de dispersión tanto horizontal como vertical de los contaminantes, causando que se acumulen en dichas zonas con grave riesgo para la salud. En las zonas afectadas por estas inversiones suele surgir niebla, que puede ser de larga duración si se trata de un valle, ya que la propia niebla impide que la superficie de la tierra se caliente, prologando la inversión.

Precipitación Las precipitaciones sirven como proceso de limpieza de contaminantes en la atmósfera de tres maneras: (i) arrancando partículas grandes en la caída de gotas o nieve, (ii) aglomerando partículas pequeñas en las nubes formando gotas o nieve y (iii) eliminando contaminantes gaseosos por disolución o absorción.

Topografía Otro factor que afecta a la meteorología y por tanto indirectamente a la polución es la topografía del área de estudio. Por ejemplo, en las zonas de valles, como ya se ha dicho, se producen grandes inversiones, sobre todo en invierno. En zonas industrializadas y pobladas, la parte baja de los valles son especialmente críticas.

Respecto a las ciudades, el terreno endurecido por los edificios aumenta la turbulencia del aire, aumentando la dispersión de contaminantes, sin embargo al mismo tiempo los edificios y el asfalto almacenan calor recibido durante el día, lo que sumado al calentamiento causado por las calefacciones durante los meses fríos, crea un diferencial de temperatura y presión entre la ciudad y áreas rurales colindantes, aumentando la circulación hacia el interior de la ciudad, concentrando los contaminantes.

2.1.3. Recogida de datos

En España el control de la calidad del aire se lleva a cabo mediante un conjunto de redes de medición de datos de calidad del aire, con las Comunidades Autónomas y Entidades Locales teniendo la competencia de su gestión.

En el caso de Valladolid, se cuenta con una estación gestionada por la Junta de Castilla y León, situada en Medina del Campo, mientras que el resto de estaciones corresponden a la red del Ayuntamiento de Valladolid y empresas privadas. La Red de Control de la Contaminación Atmosférica del Ayuntamiento de Valladolid (RCCAVA) está formada por cinco estaciones de medida, cada una recogiendo datos de un subconjunto de los contaminantes vistos anteriormente. Sumado a estas hay cinco estaciones gestionadas por empresas privadas situadas dentro de la provincia de Valladolid [15].

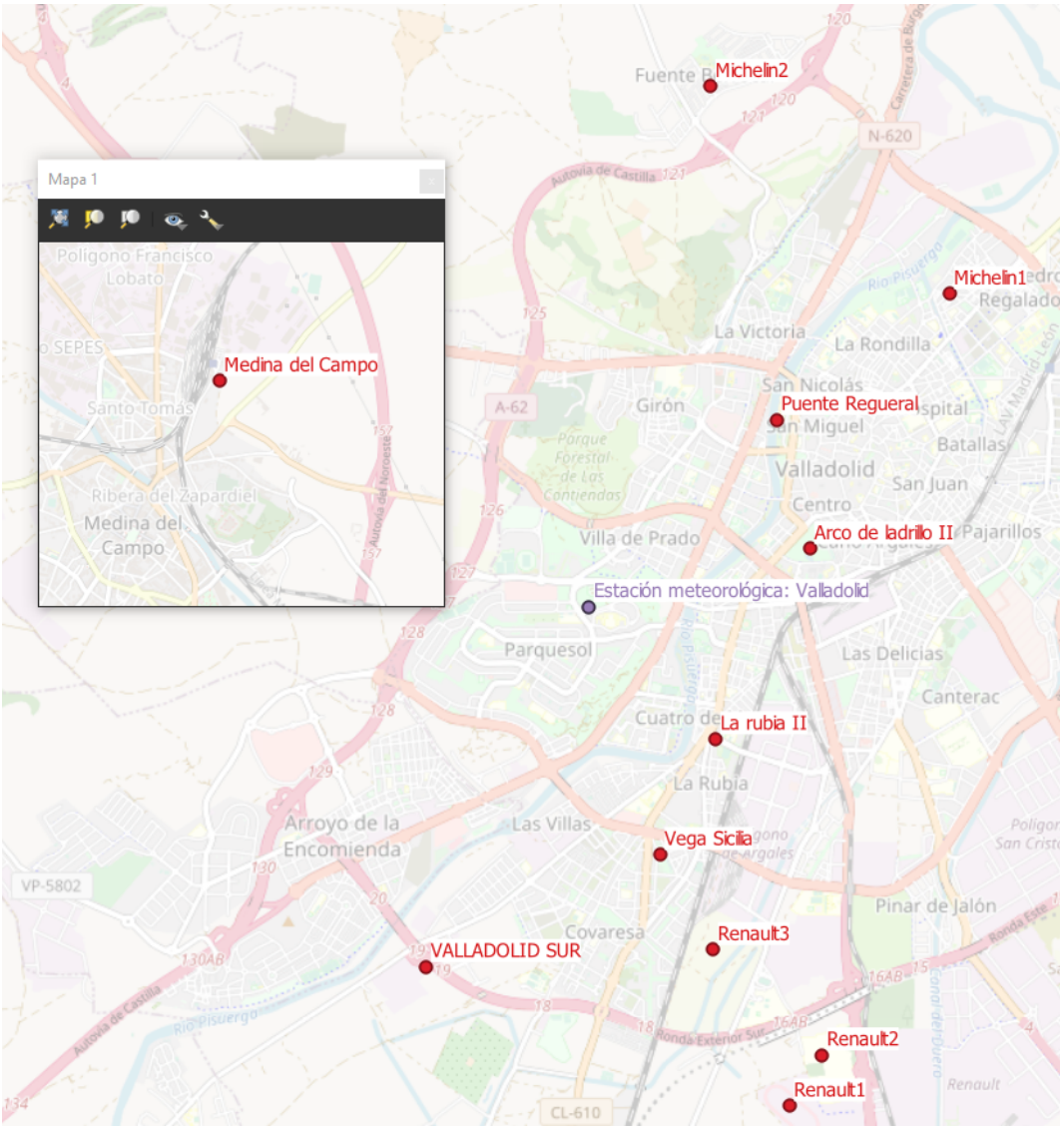


Figura 2.1: Situación de las estaciones de recogida de datos en Valladolid - Fuente: elaboración propia

Estación	SO2	PM10/PM2,5	NO/NO2	CO	O3
Arco de Ladrillo		✓	✓	✓	
La Rubia	✓	✓	✓		
Vega Sicilia		✓	✓		✓
Puente del Poniente		✓	✓		✓
Valladolid Sur			✓		✓
Medina del Campo*	✓	Sólo PM10	✓		✓
Michelin1**			✓		✓
Michelin2**			✓		✓
Renault1**			✓		✓
Renault2**		Sólo PM10	✓	✓	
Renault3**		Sólo PM10	✓		

* Estación gestionada por la Junta de Castilla y León, fuera de RCCAVA
** Estaciones gestionadas por empresas privadas, fuera de RCCAVA

Tabla 2.1: Contaminantes medidos en cada estación de Valladolid

Aunque muchas de estas estaciones no están gestionadas directamente por la Junta de Castilla y León, cabe destacar que los datos de todas son accesibles a través del portal de Control de Calidad del Aire de la Junta de Castilla y León[1], y sus datos completos aparecen en los inventarios de estaciones ofrecidos por este organismo.

2.2. Big Data en el control de la calidad del aire

Además del esfuerzo por parte de gobiernos e instituciones públicas para ofrecer herramientas de control de la calidad del aire, a lo largo del tiempo han surgido múltiples alternativas no oficiales para la obtención, visualización y predicción de la polución de una zona, algunas de ellas sin ánimo de lucro y otras de carácter comercial.

En esta sección exploramos algunas de las herramientas más completas disponibles con un objetivo común: ofrecer a sus usuarios información sobre la calidad del aire en cierto momento - normalmente en tiempo real - en una zona determinada. Ambas herramientas estudiadas en esta sección explotan los datos mediante visualizaciones y predicción.

Comenzamos con una breve exploración de un portal elaborado por un organismo oficial: el Índice de Calidad del Aire Europeo, de la Agencia Europea de Medio Ambiente. Respecto a proyectos no oficiales, consideraremos BreezoMeter.

2.2.1. Índice de Calidad del Aire Europea

La Agencia Europea del Medio Ambiente ofrece múltiples recursos relacionados con calidad del aire, entre ellos estudios de interés como el impacto de medidas contra la COVID-19 en la concentración de contaminantes en el

[1] http://servicios.jcyl.es/esco/index.action. Visitado por última vez el 18 de Septiembre de 2020.

aire[2]. Una de estas visualizaciones es un mapa interactivo con el índice de calidad del aire Europeo para múltiples ciudades de toda Europa, incluyendo una explicación de cómo se elabora dicho índice.

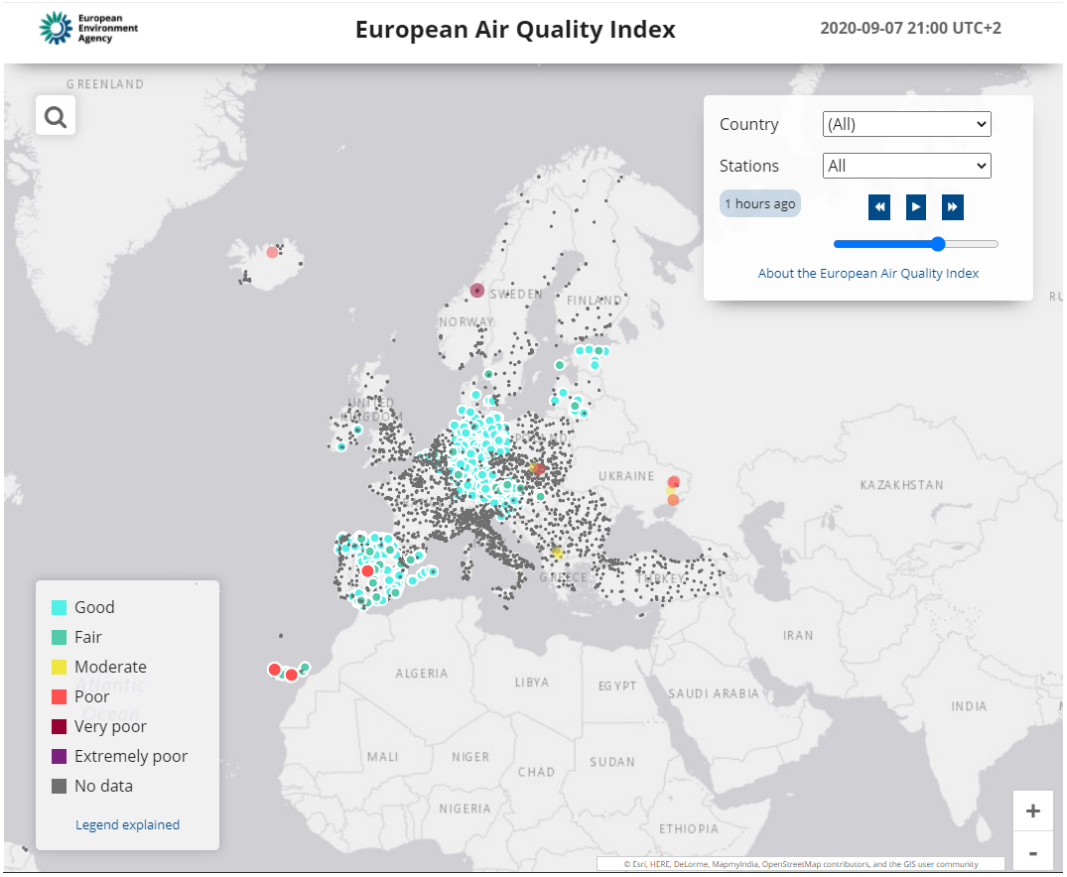


Figura 2.2: Mapa interactivo del Índice de Calidad del Aire Europeo, mediciones para una hora antes - Fuente: https://www.eea.europa.eu/themes/air/air-quality-index/index

[2]https://www.eea.europa.eu/themes/air/air-quality-and-covid19. Visitado por última vez el 18 de Septiembre de 2020

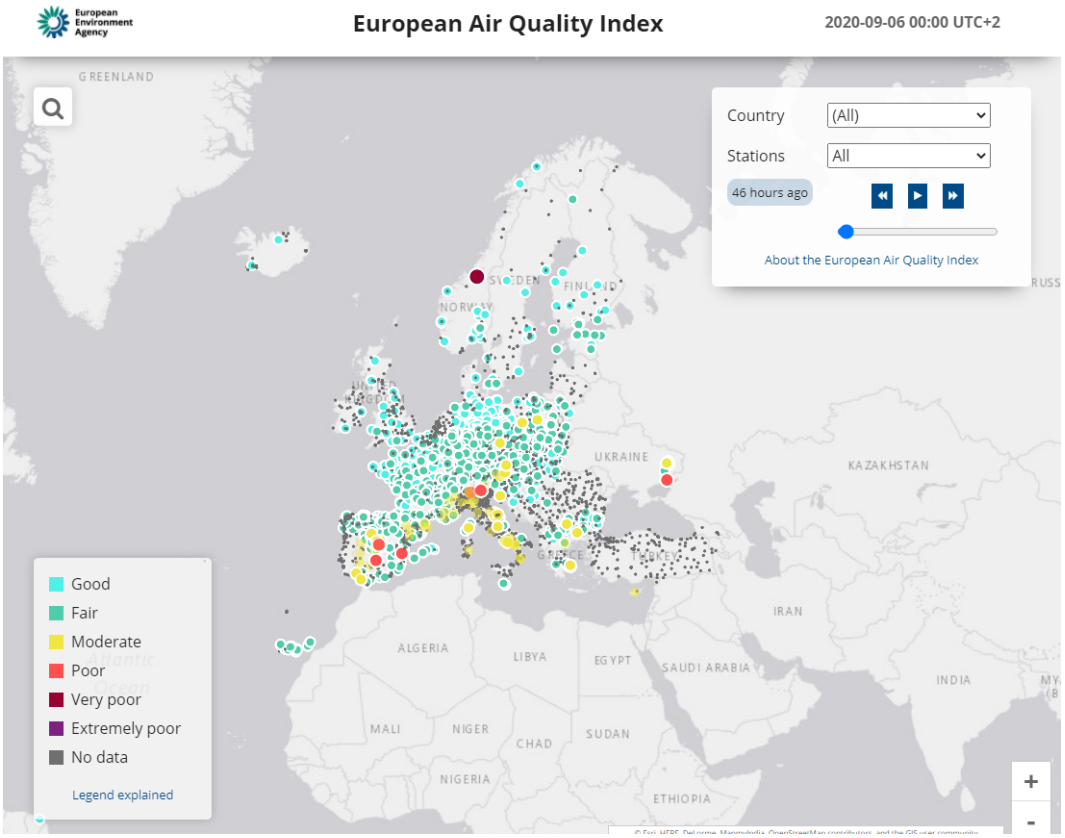


Figura 2.3: Mapa interactivo del Índice de Calidad del Aire Europeo, mediciones para 46 horas antes - Fuente: https://www.eea.europa.eu/themes/air/air-quality-index/index

Como se puede ver, aún con la cantidad masiva de estaciones disponibles en este portal, la mayoría no parecen tener datos. Cambiando la temporalidad a 46 horas antes de la hora actual, la mayoría de las estaciones tienen datos, aunque sigue habiendo una ausencia para algunas de ellas, especialmente en el sureste de Europa. Pinchando sobre cada estación podemos ver una vista detallada de los valores medidos en la estación, así como recomendaciones a la población del área:

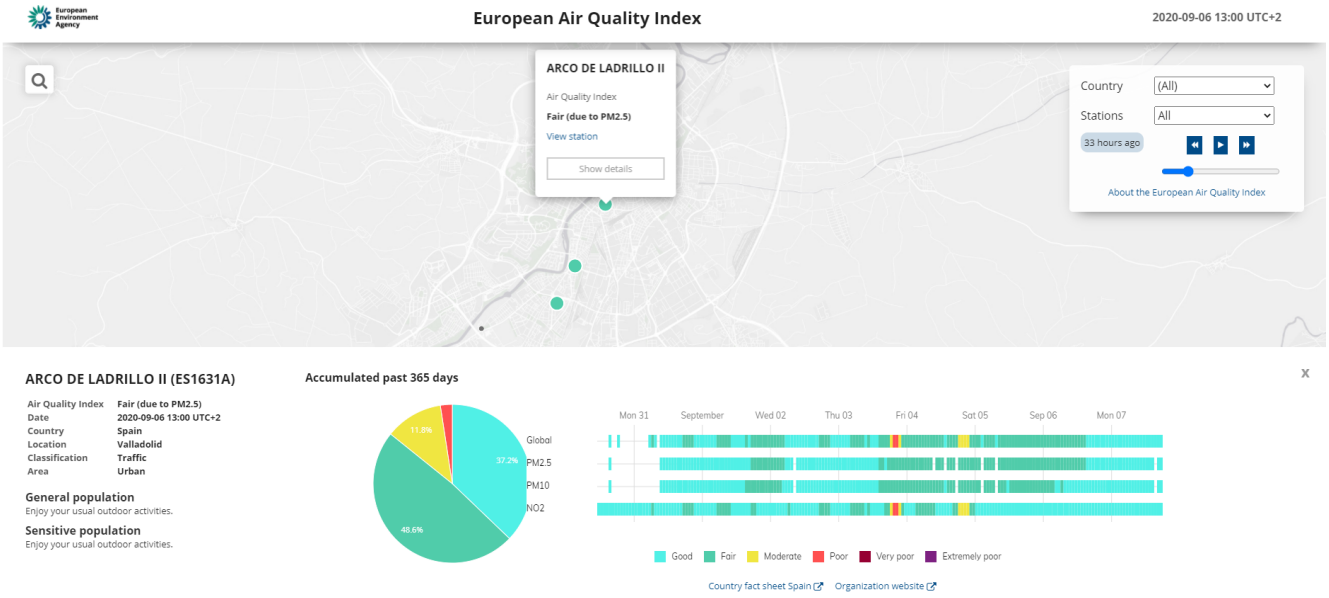


Figura 2.4: Detalles de la estación Arco de Ladrillo II según el Índice de Calidad del Aire Europeo - Fuente: https://www.eea.europa.eu/themes/air/air-quality-index/index

Esta herramienta también ofrece predicciones hasta un día por delante de la hora actual:

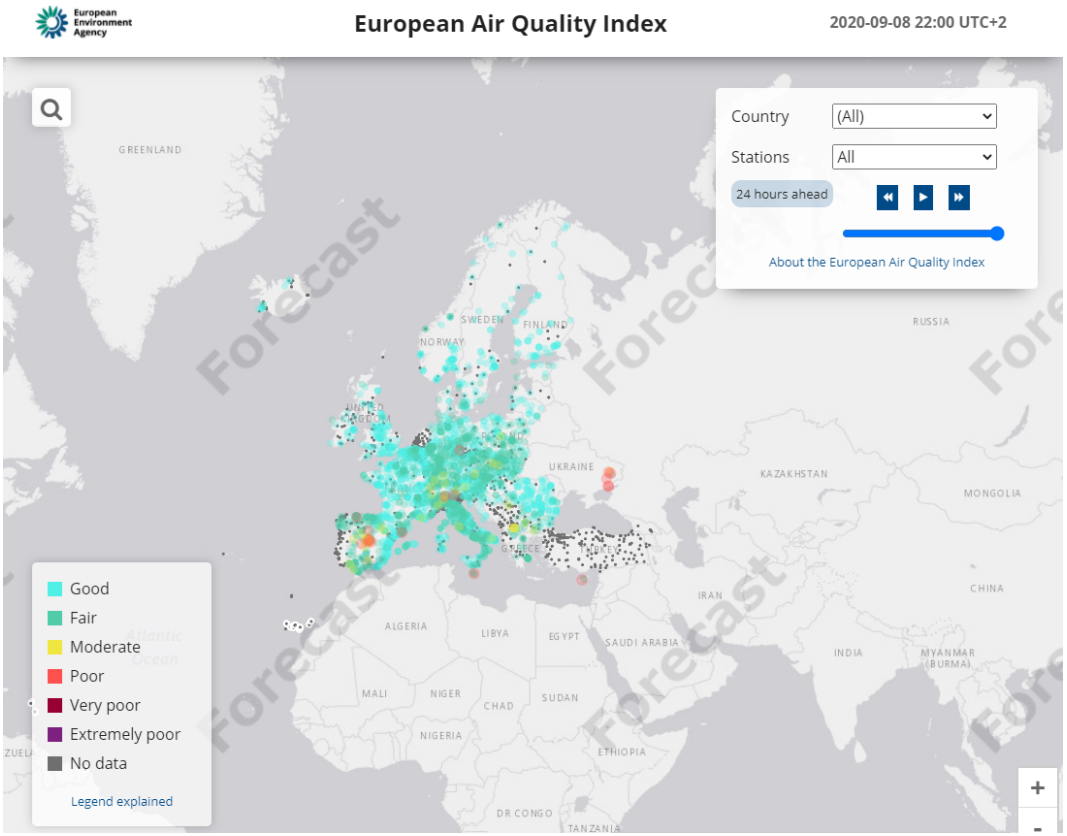


Figura 2.5: Predicción 24 horas por delante de la hora actual según el Índice de Calidad del Aire Europeo - Fuente: https://www.eea.europa.eu/themes/air/air-quality-index/index

Estas predicciones se basan en el modelo de *ensembles* de CAMS[3] (Copernicus Atmosphere Monitoring Service), una red de monitorización de la atmósfera, que, entre otros productos, ofrece una predicción de la calidad del aire en Europa basado en nueve modelos numéricos de predicción de contaminantes [16, 17].

2.2.2. BreezoMeter

BreezoMeter es una compañía creada en 2012 y dedicada a ofrecer información sobre la calidad del aire en tiempo real y con una resolución geográfica con una granularidad de unos 500 metros. Son capaces de alcanzar esta resolución geo-temporal utilizando múltiples datasets (sensores, satélites, meteorología, transporte, CAMS, etc.) para modelar la dispersión de la polución [2].

[3]https://www.regional.atmosphere.copernicus.eu/. Visitado por última vez el 18 de Septiembre de 2020

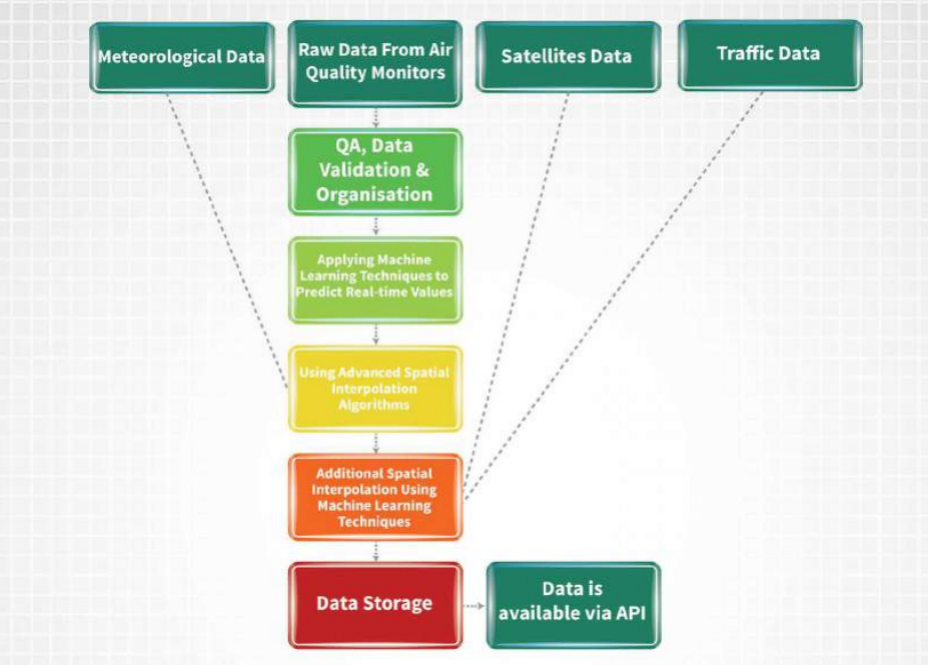


Figura 2.6: Flujo de datos de la solución de BreezoMeter - Fuente: One, Two, Three, Breathe [2]

Su producto final es una API que ofrece, bien de manera gratuita y limitada o de pago con menos limitaciones, un índice de calidad global, un mapa de calor de polución para una zona, la concentración de contaminantes, recomendaciones teniendo en cuenta el estado actual, así como un histórico de datos y predicción de 24 a 96 horas.

En su portal podemos ver una visualización del tipo de datos ofrecidos, con un mapa de calor de la contaminación en Europa:

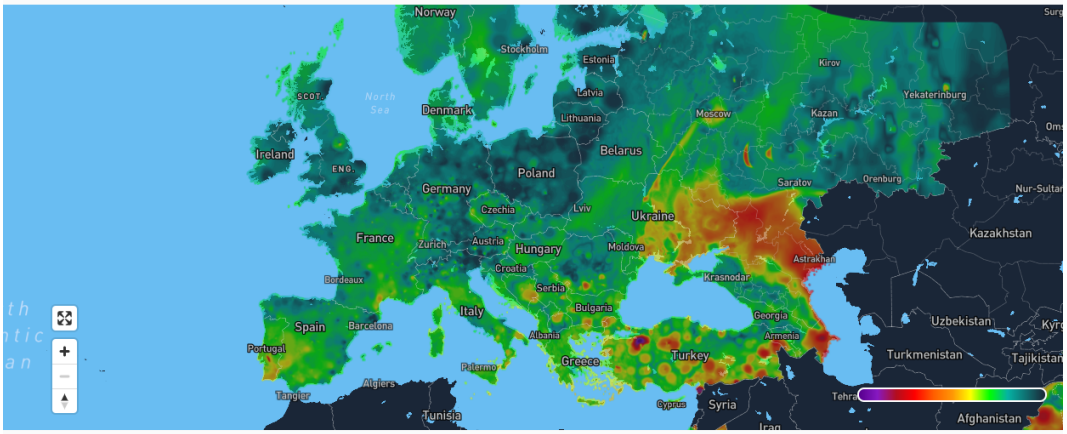


Figura 2.7: Mapa de calor de contaminación en Europa según BreezoMeter - Fuente: https://breezometer.com/air-quality-map

Y los valores medidos a nivel de calle, así como predicciones, recomendaciones e información sobre los contaminantes existentes.

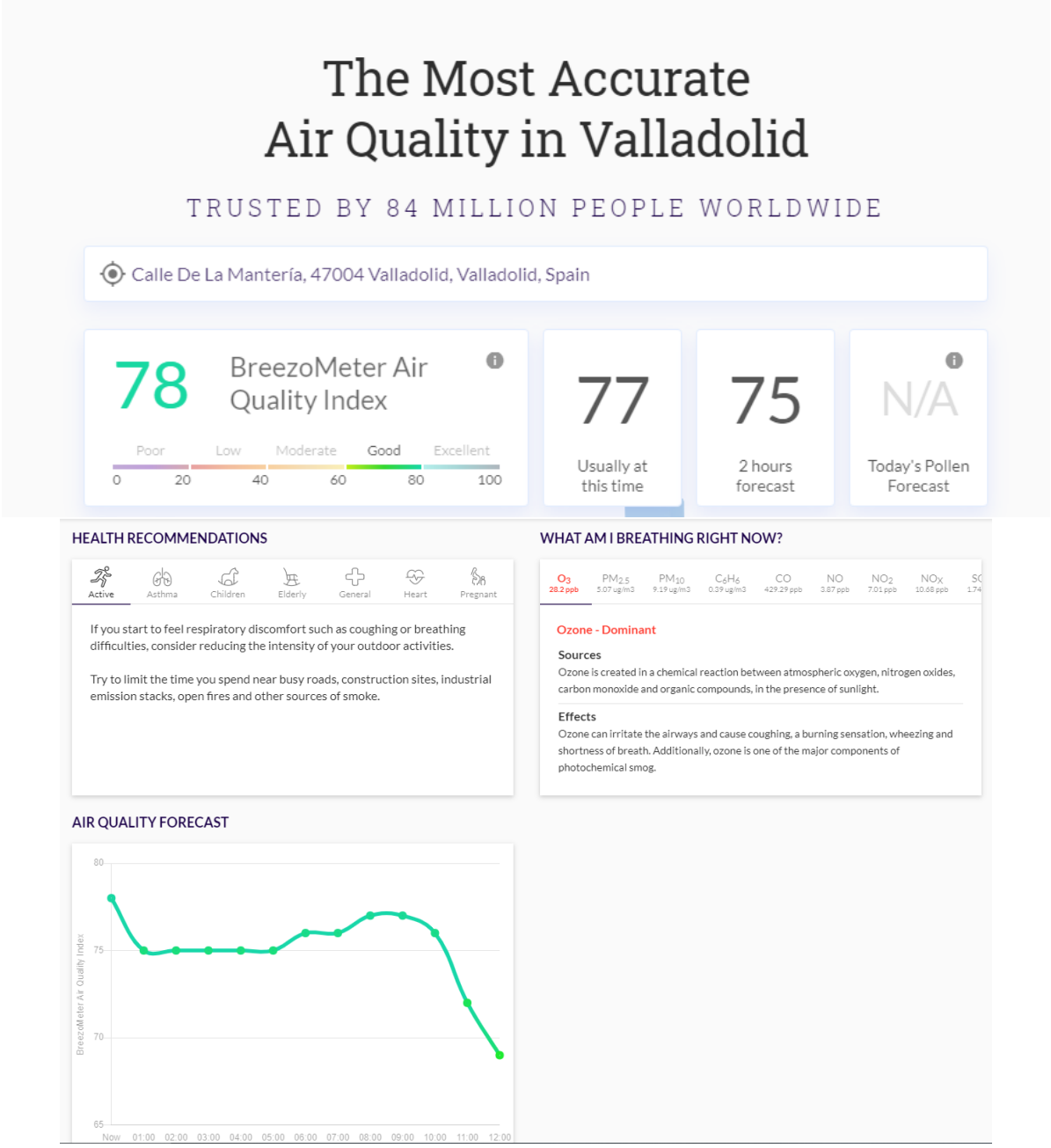


Figura 2.8: Estado actual del aire en Valladolid según BreezoMeter - Fuente: https://breezometer.com/air-quality-map

Al tratarse de un producto comercial, se desconocen los detalles del algoritmo utilizado para la predicción,

aunque como ya se ha mencionado utilizan un amplio rango de fuentes de datos. BreezoMeter ofrece algunos resultados en términos del error cuadrático medio (RMSE) para cada contaminante que predicen, que consideran satisfactorios:

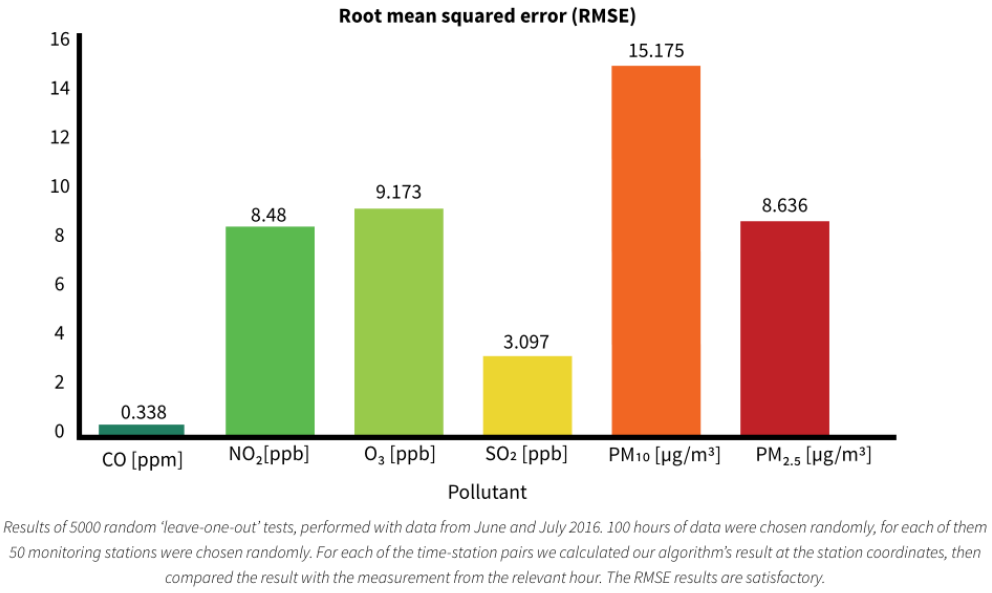


Figura 2.9: RMSE por contaminante resultante de 5000 tests aleatorios en datos de Junio y Julio de 2016 - Fuente: One, Two, Three, Breathe [2]

2.3. Técnicas de aprendizaje automático

Para finalizar esta sección, se presenta el fundamento teórico sobre el que se apoyan los algoritmos utilizados en este trabajo para la predicción de la calidad del aire (Sección 7.2). Mientras que en la literatura se utilizan algoritmos complejos como redes neuronales y otras técnicas de Deep Learning, este trabajo se realiza bajo la limitación de tratarse de una aplicación Big Data diseñada para manejar enormes volúmenes de datos, mientras que los algoritmos con mejor resultado comprobado (por ejemplo, las redes neuronales) serían demasiado lentas sin contar con entornos optimizados para su aplicación. Aunque existen ejemplos en la literatura de implementación de redes neuronales con el paradigma Map Reduce [18, 19], el producto que permite realizar experimentos de Deep Learning en Hadoop, Apache Submarine[4], aún se encuentra en sus primeras versiones. Por tanto, debido a su menor complejidad y su disponibilidad en la librería *ML* de *Spark* se han utilizado los siguientes modelos: (i) Regresión Lineal, (ii) Árbol de decisión, y (iii) Random Forest.

También se dedica un primer apartado a explicar la metodología llevada a cabo para el entrenamiento de los modelos, común a los tres algoritmos mencionados, y una breve comparativa entre los modelos. Las fuentes principales de esta sección son *Statistics for Machine Learning* de Pratap Dangeti [20] y *Comparative Study on*

[4] https://submarine.apache.org/ Visitado por última vez el 18 de Septiembre de 2020

Classic Machine Learning Algorithms de Danny Varghese [21].

2.3.1. Entrenamiento

Dado que el objetivo de la tarea de aprendizaje automático es la de utilizar datos pasados para crear modelos matemáticos que puedan predecir el futuro, es necesario dividir el conjunto de datos en dos: uno para entrenamiento y validación, que será el que servirá para ajustar el modelo a la realidad, y uno de test, con datos que no se hayan utilizado previamente, lo que nos permite comprobar la eficacia del modelo en un contexto de predicción de datos futuros. Esta división se hace aleatoriamente reservando un tercio de los datos para test y dos tercios para entrenamiento y validación.

Una de las características de los tres algoritmos utilizados es que todos tienen parámetros que deben ser elegidos cuidadosamente para obtener el mejor modelo. Estos parámetros han sido seleccionados utilizando una técnica conocida como Validación Cruzada de 5 particiones, por la cual para cada conjunto de parámetros:

1. El conjunto de entrenamiento y validación se divide aleatoriamente de nuevo en 5 partes, $4/5$ se designan para entrenamiento, y el quinto restante como conjunto de validación.
2. Se entrena un modelo con los parámetros actuales usando los datos de entrenamiento.
3. Se calcula una medida de la calidad del modelo (en nuestro caso, el error cuadrático medio (RMSE)) sobre los datos de validación. Este error cuadrático medio se define como:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y} - Y)^2}{n}} \tag{2.1}$$

Donde $\hat{Y}$ es el valor predicho, Y es el valor real de la variable, y n el número de observaciones en el conjunto de datos de validación.

4. Se vuelven a elegir los datos para validación y entrenamiento, siempre tomando datos de validación que no se hayan utilizado previamente para esta tarea. Se vuelve a iterar desde el paso 2.

Una vez se han utilizado todos los datos para validación una vez, se calcula la media del RMSE obtenido en los cinco experimentos, repitiendo todos los pasos para todos los conjuntos de parámetros existentes. Finalmente se eligen los parámetros con el mínimo RMSE medio, y ese será el modelo que se evaluará con el conjunto de test, para obtener una medida de la bondad del modelo.

Los modelos resultado del entrenamiento se evalúan y comparan según la métrica del R^2 ajustado, que es una versión mejorada del R^2 que penaliza al modelo si utiliza demasiadas variables. Esta métrica se define como:

$$R^2 ajustado = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1} \tag{2.2}$$

Donde N es el número de observaciones en el conjunto de test, p es el número de variables utilizadas en el modelo, y R^2 es una medida de la cantidad de variación en la variable predicha (Y) que es explicada por la variación en las variables predictoras (X), se define como:

$$R^2 = 1 - \frac{\sum(Y - \hat{Y})^2}{\sum(Y - \bar{Y})^2} \tag{2.3}$$

Donde Y es el valor real de la variable a predecir, $\hat{Y}$ es el valor predicho, y $\bar{Y}$ es la media de Y sobre el conjunto de datos.

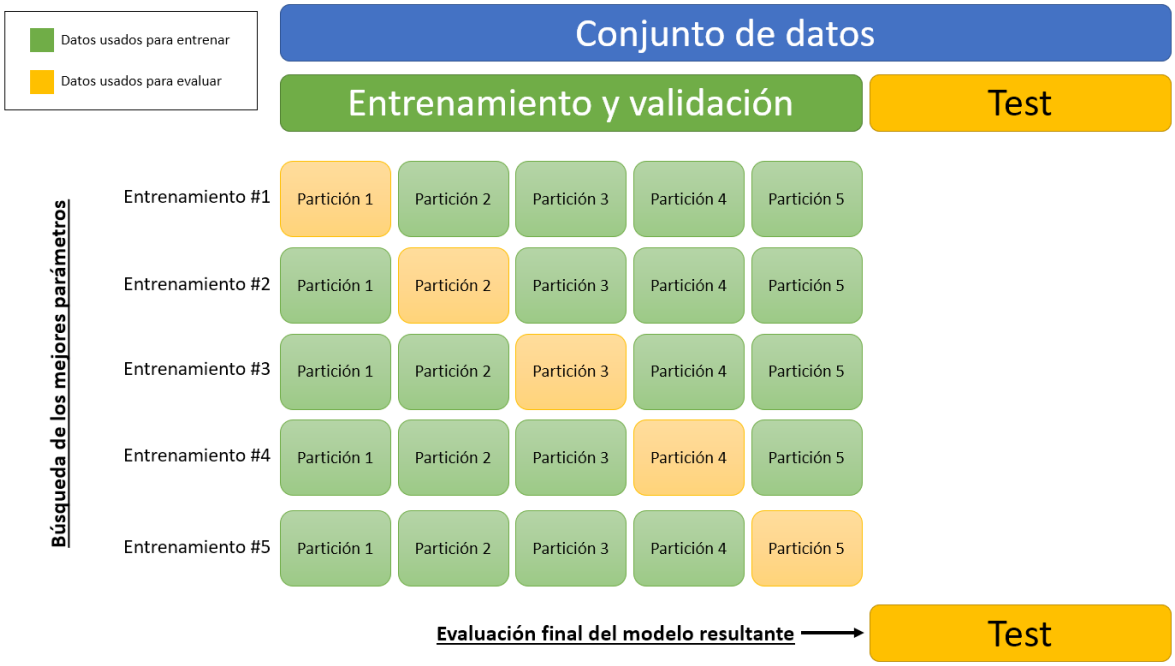


Figura 2.10: Esquema del particionado de datos para el entrenamiento - Fuente: elaboración propia

2.3.2. Regresión lineal

Una regresión o ajuste lineal es un modelo que aproxima una variable Y a partir de una o más variables regresoras X cuyo valor conocemos, suponiendo una relación lineal entre las variables X e Y. Un modelo de regresión lineal con k variables regresoras es una ecuación del tipo:

$$Y = \alpha_0 + \beta_1 X_1 + ... + \beta_k X_k + \varepsilon \tag{2.4}$$

Donde α_0 es el término independiente, que define el valor de Y cuando todas las variables regresoras son 0. β_i es el coeficiente de la variable X_i, y representa cuanto aumenta (o disminuye si negativa) Y cuando X_i aumenta en

una unidad. ε es una variable aleatoria con media 0 y distribución normal que representa el error aleatorio cometido al predecir Y.

Un modelo de regresión lineal se entrena buscando valores para β y α_0 que minimicen el error cuadrático medio (RMSE) en un conjunto de observaciones para las cuales sabemos el valor real de Y.

Regularización La regularización en un modelo de regresión lineal es una técnica utilizada para realizar selección de variables, así como evitar que el modelo sobre-entrene los datos, causando malos resultados en los datos de test.

Existen tres tipos: Regularización L1 (Regresión Lasso), L2 (Regresión Ridge) o ElasticNet, que aplica ambas regularizaciones. Las regularizaciones L1 y L2 añaden un término de penalización a la función a minimizar durante el entrenamiento (en nuestro caso el error cuadrático), así la Regresión Lasso busca eliminar los coeficientes con poca relevancia en el modelo, y la Regresión Ridge minimizar la magnitud de coeficientes.

Matemáticamente se definen como:

$$errorL1 = (Y - \hat{Y})^2 + \lambda \sum_{j=1}^{k} |\beta_j| \tag{2.5}$$

$$errorL2 = (Y - \hat{Y})^2 + \lambda \sum_{j=1}^{k} \beta_j^2 \tag{2.6}$$

$$errorElasticNet = (Y - \hat{Y})^2 + r \cdot \lambda \sum_{j=1}^{k} |\beta_j| + (1 - r) \cdot \lambda \sum_{j=1}^{k} \beta_j^2 \tag{2.7}$$

Todas las regularizaciones tienen un parámetro λ que controla la fuerza de la regularización, cuando $\lambda = 0$ la penalización no tiene impacto, mientras que un parámetro que tiende a infinito causará que los coeficientes se hagan cero. Además, ElasticNet tiene un parámetro r entre 0 y 1 que controla cual de las dos regularizaciones, L1 o L2, tiene más fuerza. En 0, se trataría de una regresión Ridge, mientras que en 1 se trata de una regresión Lasso.

2.3.3. Árbol de decisión

Un árbol de decisión es un modelo estadístico que utiliza reglas aplicadas sobre las variables predictoras X para predecir Y. Cada nodo intermedio del árbol representa una regla, y genera dos ramas, una donde la regla se cumple, y otra donde no. Estas reglas tienen la forma: $X = valor$ para variables discretas o $X > umbral$ para variables numéricas. Los nodos hoja contienen la predicción para aquellos datos que hayan seguido el camino que lleva a ellos desde el nodo raíz.

La decisión sobre qué variables seleccionar y los valores o umbrales utilizados para elaborar las reglas se toma tratando de minimizar una función, generando reglas hasta que se alcanza algún tipo de criterio (por ejemplo altura máxima del árbol).

Un ejemplo de un árbol de decisión sería el siguiente, que utiliza las características de los pasajeros del Titanic para predecir si sobrevivieron al accidente o murieron:

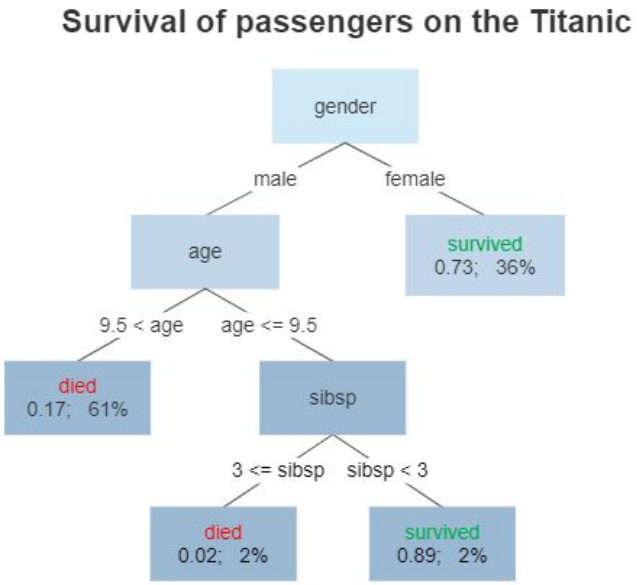


Figura 2.11: Ejemplo de árbol de decisión - Fuente: Gilgoldm / CC BY-SA

Este modelo es una representación de un tipo de regresión no lineal, en la que cada regla genera un plano en el espacio que separa los datos, con el árbol completo separando los datos en conjuntos con la misma predicción. Un ejemplo de un árbol de decisión y su representación en el plano es el siguiente, donde se trata de predecir si existe una deformación de la columna entre niños tras haberles operado, utilizando su edad y la primera vértebra operada como variables predictoras. Los nodos hoja en este caso contienen la probabilidad de que la deformación esté presente en niños con dichas características:

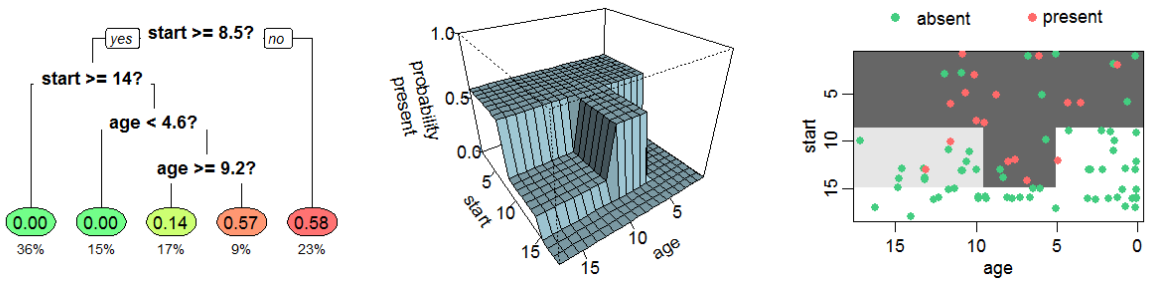


Figura 2.12: Ejemplo de árbol de decisión - Fuente: Stephen Milborrow / CC BY-SA

Mientras que ambos son ejemplos de clasificación binaria (con etiquetas sobrevive/muere y presente/ausente respectivamente), es posible utilizar estos algoritmos para realizar regresiones, en cuyo caso la función a minimizar durante el entrenamiento es RMSE.

2.3.4. Random Forest

Un modelo Random Forest (traducido literalmente, bosque aleatorio), es un algoritmo de tipo *ensemble*, así llamados porque se generan entrenando múltiples modelos de un tipo, y seleccionando la predicción media de todos ellos. Random Forest se creó para tratar de solventar las carencias de los árboles de decisión (principalmente el hecho de que son demasiado sensibles a pequeños cambios en los datos), entrenando múltiples árboles de decisión, reduciendo la sensibilidad.

Random Forest muestrea las observaciones y variables a la hora de crear árboles, buscando independencia entre ellos. En tareas de clasificación, la predicción resultante es en la que coinciden la mayoría de árboles, en regresión se trata de la media de las predicciones.

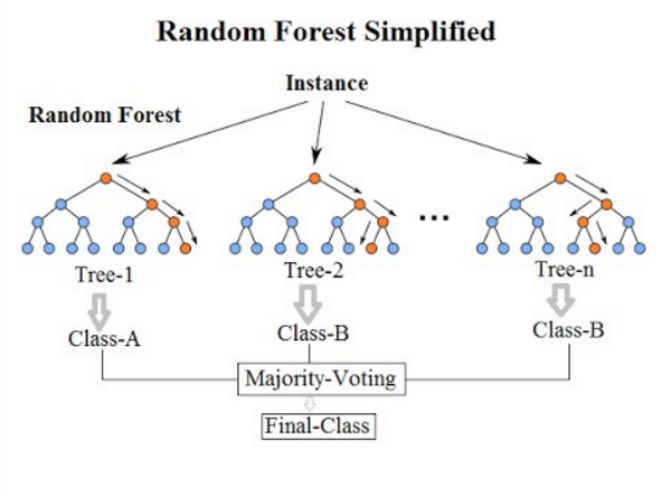


Figura 2.13: Ejemplo de Random Forest - Fuente: Venkata Jagannath / CC BY-SA

2.3.5. Comparativa entre algoritmos

Por último, se incluye un resumen de las ventajas e inconvenientes de cada método explorado:

Regresión lineal

- Con un número bajo de variables regresoras el entrenamiento es rápido, con una complejidad cúbica respecto al número de variables regresoras, y lineal respecto al número de instancias de entrenamiento:$\mathcal{O}(k^2n + k^3)$.

- Resultados fácilmente interpretables, ya que el coeficiente β_i es asimilable a la significación de la variable X_i.
- Asume múltiples hipótesis sobre los datos, las relaciones entre variables, y el error resultante de la evaluación del modelo que, en caso de no cumplirse, hacen al modelo inválido.
- Sólo permite soluciones lineales.
- Sólo permite variables continuas.
- Es muy sensible a valores extremos, cuya presencia puede deteriorar significativamente la capacidad predictiva del modelo.

Árbol de decisión

- La complejidad es más alta que en la Regresión Lineal (asumiendo que el número de observaciones es mucho mayor que el número de variables), ya que el modelo debe iterar por cada posible regla, lo que implica iterar por cada variable (k) y cada posible valor umbral (n), calculando RMSE (complejidad $\mathcal{O}(n)$), resultando en una complejidad total de $\mathcal{O}(n^2k)$.
- Modelo de muy fácil interpretabilidad y visualización.
- No asumen ningún tipo de hipótesis sobre la distribución de los datos.
- Produce soluciones no lineales.
- Puede trabajar con variables continuas mediante discretización, aunque los resultados son significativamente mejores cuando la mayoría de variables son discretas.
- Si no se selecciona el criterio de parada con cuidado, puede resultar en sobre-entrenamiento. Es muy sensible a cambios en los datos.
- Menos sensibles a valores extremos en los datos.

Random Forest

- Ya que este modelo se basa en crear N_{trees} árboles la complejidad será la del árbol de decisión multiplicada por dicho parámetro: $\mathcal{O}(N_{trees}n^2k)$.
- La interpretación y visualización se torna más complicada al tratarse de un conjunto de árboles.
- Mucho más robusto contra sobre-entrenamiento y menos sensible a cambios.

Capítulo 3

Metodología y arquitectura de la solución

En este capítulo se explica la metodología de trabajo seguida, así como las herramientas y la arquitectura utilizada para la implementación del Data Lake y el entorno de explotación.

3.1. Metodología

La metodología seguida en este trabajo es la indicada en la Figura 3.1, con un proceso iterativo cuyo objetivo es la utilización de fuentes de datos para extraer conocimientos útiles, con un proceso de mejora en el cual se evalúa cómo se ajustan los resultados a los objetivos en cada etapa del ciclo. Más detalladamente los pasos del proceso son:

- **Inicio:** En esta fase se establecen los objetivos del proyecto.
- **Análisis**: En este momento se establecen los requisitos según los objetivos del proyecto, seguido de la caracterización del modelo conceptual, reflejado en la sección 4.2, se buscan fuentes de datos necesarias para la consecución de los requisitos y se diseña aquellos pasos necesarios para convertir las fuentes de datos brutas en el modelo conceptual.
- **ETL**: Es la implementación de lo planteado anteriormente, llevando a cabo la ingesta de los datos en bruto, su transformación en datos enriquecidos, y su carga en el sistema de explotación. Esta fase se sirve del Data Lake como sistema de almacenamiento y procesamiento, hasta la exportación de los datos refinados a una base de datos relacional.
- **Evaluación**: Los datos resultantes del proceso ETL se someten a una evaluación mediante las herramientas de explotación elaboradas, donde se comprueba que se cumplen los objetivos determinados inicialmente, comenzando un nuevo ciclo con el conocimiento adquirido durante esta fase.
- **Despliegue:** Una vez comprobamos que se cumplen los objetivos marcados, la solución se incorpora al producto final.

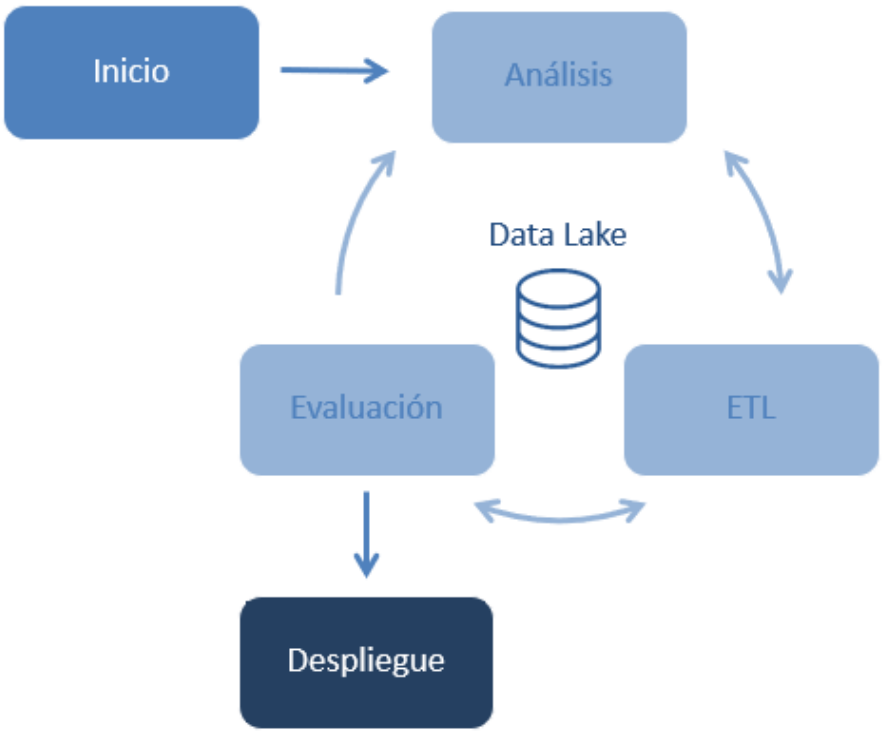


Figura 3.1: Metodología del proyecto

3.2. Data Lake

En la última década, el volumen de datos almacenados y usados por las organizaciones ha ido aumentando de manera acelerada, por lo que está cada vez más afianzado el uso de herramientas Big Data, con el Data Lake como elemento central de su arquitectura. Un Data Lake permite almacenar grandes volúmenes de datos en su forma original, retrasando la necesidad de estructurar y procesar dichos datos hasta que surge una necesidad de uso. Esto es de especial importancia cuando se dispone de unas necesidades de almacenamiento para diversos casos de uso. A diferencia del Data Warehouse tradicional, que debe servir un único propósito, un Data Lake permite a sus usuarios disponer de un repositorio centralizado de datos que puede servir diversas necesidades y proyectos [22, 23].

Para este trabajo la arquitectura Data Lake permite gestionar una gran cantidad de datos, que además puede soportar un crecimiento exponencial que se produciría, por ejemplo, al considerar todo Castilla y León en lugar de sólo Valladolid. También nos proporciona una cohesión completa con otras herramientas del entorno Hadoop, muy útiles para tareas de automatización y procesado de grandes volúmenes de datos.

La arquitectura del Data Lake propuesta es la siguiente, formada por múltiples herramientas, siete componentes, y dos partes diferenciadas: el proceso ETL llevado a cabo en un clúster Hadoop; y el entorno de explotación, formado por una base de datos relacional y software ejecutándose sobre una sola máquina virtual:

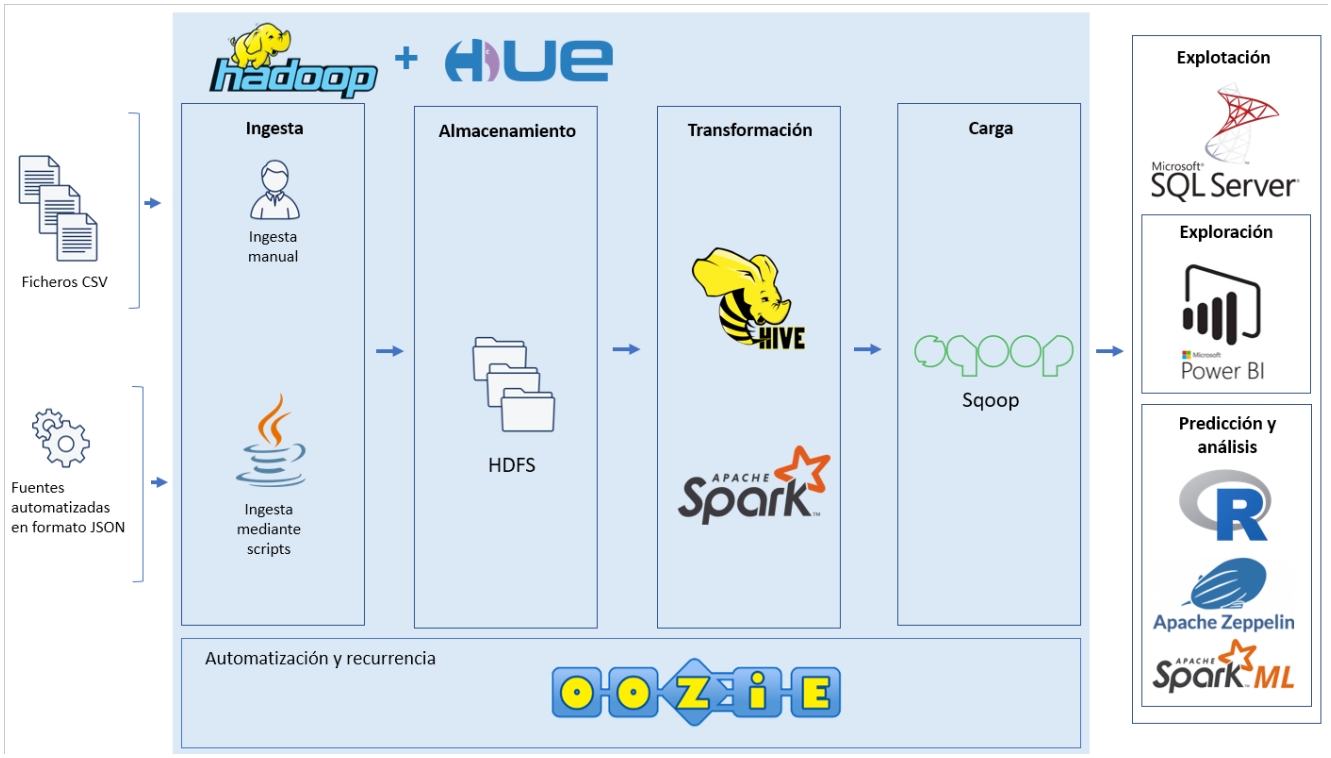


Figura 3.2: Esquema de la arquitectura y herramientas utilizadas en el proyecto - Fuente: elaboración propia

Gran parte de la arquitectura propuesta se ejecuta dentro del entorno Hadoop, utilizando Hue como la interfaz que permite el acceso a los datos y herramientas. Respecto al sistema subyacente, el proyecto se ha ejecutado en su totalidad mediante recursos y hardware proporcionado por el Departamento de Informática de la Universidad de Valladolid. Más concretamente Hadoop se ejecuta en un cluster Cloudera con nodos virtualizados a través de Proxmox, mientras que el entorno de explotación se ejecuta en una máquina virtual que describiremos en más detalle en el siguiente apartado.

Los componentes que forman nuestra arquitectura son:

- **Fuentes de datos e Ingesta:** Disponemos de dos tipos de orígenes de datos, ficheros CSV y fuentes automatizadas en formato JSON. Hablaremos con más detenimiento de su contenido en la sección 4.1 y de los detalles de su Ingesta en la sección 5.1.1. Las herramientas utilizadas para ingestar estos datos han sido una ingesta manual a través de la interfaz de Hue, que permite la subida de ficheros para su almacenaje en HDFS, y la ingesta mediante scripts, en nuestro caso escritos en Java, soportado de manera nativa por Hadoop.

- **Almacenamiento:** El sistema de almacenamiento en Hadoop es HDFS, un sistema de ficheros distribuido diseñado para ser ejecutado en hardware de bajo coste. Está construido para almacenar grandes volúmenes de datos sin penalizaciones en su velocidad, además de tener incluidas importantes medidas para la protección frente a fallos de hardware [24].
- **Transformación**: Las tareas de transformación de los datos se han llevado a cabo utilizando tanto Apache Hive como Spark. Hive permite la lectura y escritura de ficheros HDFS a través de una interfaz de lenguaje SQL [25], mientras que Spark presenta una herramienta mucho más generalista que permite ejecutar procesos Map Reduce a través de múltiples lenguajes como Java, Scala, Python y R [26]. En nuestro caso se ha utilizado la variante de Python.
- **Carga**: La transferencia de los datos del cluster Hadoop al entorno de explotación se realiza mediante Apache Sqoop, que permite transferencias de datos bidireccionales desde Hadoop a almacenes estructurados, como es el caso del Microsoft SQL Server utilizado en este trabajo [27].
- **Automatización y recurrencia:** Dado que el flujo ETL debe ser ejecutado diariamente, es importante disponer de una herramienta de automatización y programación de flujos de datos integrada con el Data Lake. Para ello se utiliza Apache Oozie, que soporta múltiples tipos de trabajos Hadoop y permite elaborar grafos de ejecución (workflows) que pueden ser programados en el tiempo [28].
- **Explotación**: Entorno en el que los datos que han pasado por el proceso ETL se utilizan para los propósitos marcados en los objetivos del proyecto, específicamente la exploración mediante una visualización y el análisis predictivo. Este entorno no es parte del Data Lake, por lo que se explica con mayor profundidad en la siguiente sección.

3.3. Entorno de explotación

Mientras que, por su descripción, un Data Lake es flexible a cualquier uso que se le quiera dar a los datos, esto tiene una penalización en el rendimiento respecto a sistemas relacionales cuando se quiere utilizar para un caso de uso específico. Por tanto, se ha desplegado un entorno de explotación optimizado para los dos casos de uso presentados en los objetivos del proyecto: realizar una visualización de los datos y un análisis predictivo de los contaminantes.

Este entorno se ejecuta sobre una máquina virtual del Departamento de Informática de la Universidad de Valladolid, creada específicamente para este trabajo, con las siguientes características:

- 8GB de memoria RAM.
- 4 núcleos de procesador.
- 50GB de disco duro.
- Sistema Operativo Windows 10.

En ella se ejecuta una instancia de Microsoft SQL Server 2012, en la que se ha creado una base de datos TFM con las tablas *EstacionCalAire, MedicionCalAire, SuperacionUmbral, Umbral* y *PrediccionCalAire*.

Como herramienta para la visualización se utiliza Power BI, también de Microsoft, que permite la creación de visualizaciones dinámicas con conector nativo con SQL Server. Por último para el análisis de los datos se ha utilizado el lenguaje R, especialmente diseñado para labores estadísticas y la realización de gráficos, y para la predicción se utiliza Apache Zeppelin, una herramienta que ofrece una interfaz cómoda y dinámica para tareas basadas en datos mediante «notebooks», y para procesamiento se usa de nuevo Spark, esta vez corriendo de manera local en la máquina virtual.

Dado que Zeppelin no está disponible de manera nativa en Windows, ha sido necesario descargar e instalarse un subsistema Linux, específicamente Ubuntu 18.04, a través de la Windows Store, lo que nos permite instalar y ejecutar Zeppelin, que queda accesible a través del puerto 8080 de la máquina.

Dentro de Spark, la librería elegida para el análisis predictivo es ML, ya que es más novedosa que alternativas como MLLib y tiene mejor integración con componentes como pipelines, que permiten enlazar diferentes partes de un modelo. Esta librería contiene numerosos algoritmos de aprendizaje automático así como herramientas de preprocesado de datos y evaluación de modelos [29].

Capítulo 4

Análisis y Diseño

Este capítulo presenta el trabajo de análisis y diseño del Data Lake elaborado, formado por tres partes: el descubrimiento de los datos, que incluye una descripción en detalle de cada una de las fuentes de datos utilizadas; el modelado de los datos refinados, aquel Modelo de Dominio que queremos alcanzar con el proceso ETL; y por último el diseño del Dataflow, que incluye todas las transformaciones a las que se someten a los datos en bruto para alcanzar los datos refinados, organizando cada transformación en zonas.

4.1. Descubrimiento de los datos

En esta sección se exploran las fuentes de datos elegidas, se ofrece una breve descripción de cada una, así como las características más importantes que debemos conocer para su utilización, modelado mediante perfiles.

4.1.1. Calendario laboral de Valladolid

Esta fuente contiene un listado de días festivos laborales en Valladolid, de tipo estatal, autonómico y local (en este último caso para el municipio de Valladolid), para los años 2018 a 2020. Mientras que este calendario cambia anualmente, no se considera necesario dado el alcance de este proyecto su actualización a años anteriores a 2018 o posteriores a 2020.

El origen de esta fuente son múltiples artículos de elPeriodico.com, uno por cada año recopilado[1]. De dichos artículos se ha extraído una lista de días, convertida manualmente a fechas con un formato estándar, y almacenadas

[1] Visitados por última vez el 18 de Septiembre de 2020

- **2018:** https://www.elperiodico.com/es/economia/20180328/calendario-laboral-valladolid-2018-festivos-6721802
- **2019:** https://www.elperiodico.com/es/economia/20190115/calendario-laboral-festivos-valladolid-2019-7247716
- **2020:** https://www.elperiodico.com/es/economia/20191219/calendario-laboral-festivos-valladolid-2020-7781643

en un fichero CSV.

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/Festivos*
Nombre	Un solo fichero csv *festivos20182020.csv*
Carga	Se lleva a cabo directamente a HDFS
Actualización	No se prevé actualización
Crecimiento	No se prevé crecimiento

Atributos

Índice	Nombre	Tipo	Descripción	Nullable
1.1	**fecha**	String	Fecha (formato DD-MM-YYYY) correspondiente al día festivo	NO

Tabla 4.1: Perfil de datos en bruto - Calendario laboral de Valladolid

4.1.2. Estaciones de control de calidad del aire

Esta fuente contiene un inventario de las estaciones de control de calidad del aire instaladas en Castilla y León, así como ciertos datos que ayudan a contextualizar las mediciones que recogen, como las localización geográfica o la altitud.

Estos datos se componen de dos partes, la primera de las cuales incluye los detalles de las estaciones, extraídas de un conjunto de datos actualizado anualmente por la Junta de Castilla y León[2]. No se considera necesario para el alcance del proyecto proyectar una actualización de estos datos.

La segunda parte incluye los identificadores que permiten conectar cada estación del inventario con los datos extraídos de la red de control de la calidad del aire. Se ha extraído el identificador de cada una de las estaciones existentes en la provincia de Valladolid a partir del código fuente del formulario de obtención de datos históricos en el portal de Control de Calidad del Aire de la Junta de Castilla y León[3].

[2]https://datosabiertos.jcyl.es/web/jcyl/set/es/medio-ambiente/calidad_aire_estaciones/1284212701893 Visitado por última vez el 18 de Septiembre de 2020

[3]http://servicios.jcyl.es/esco/cargarFrmDatosHistoricos.action Visitado por última vez el 18 de Septiembre de 2020

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/Estaciones/detalles*
Nombre	Un solo fichero csv *Estaciones.csv*
Carga	Se lleva a cabo directamente a HDFS
Actualización	Los datos se actualizan en la fuente anualmente, se desconoce la fecha exacta
Crecimiento	No se prevé crecimiento

Atributos				
Índice	**Nombre**	**Tipo**	**Descripción**	**Nullable**
2.1	**ESTACIÓN**	String	Nombre de la estación	NO
2.2	**LOCALIZACIÓN**	String	Localización aproximada (calle, plaza, instalación) de la estación	NO
2.3	**PROVINCIA**	String	Provincia en la que está situada la estación	NO
2.4	**LONGITUD**	String	Longitud geográfica en la que está la estación (grados, minutos, segundos y hemisferio) expresado como: Gº M' S" H	NO
2.5	**LATITUD**	String	Latitud geográfica en la que está la estación (grados, minutos, segundos y hemisferio) expresado como: Gº M' S" H	NO
2.6	**ALTITUD**	String	Altitud en metros sobre el nivel del mar al que está la estación, expresada con la altitud seguida de un espacio y la letra «m»	SI
2.7	**OPERATIVA**	String	Indica si la estación está operativa en el momento de recogida de los datos	NO

Tabla 4.2: Perfil de datos en bruto - Estaciones de control de calidad del aire (detalles)

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/Estaciones/ids*
Nombre	Un solo fichero csv *Estaciones_ids.csv*
Carga	Se lleva a cabo manualmente
Actualización	No se prevé actualización
Crecimiento	No se prevé crecimiento

Atributos				
Índice	**Nombre**	**Tipo**	**Descripción**	**Nullable**
2.8	**ESTACIÓN**	String	Nombre de la estación	NO
2.9	**id**	Integer	Identificador de la estación en la base de datos de ESCO	NO

Tabla 4.3: Perfil de datos en bruto - Estaciones de control de calidad del aire (identificadores)

4.1.3. Mediciones de calidad del aire

Se recogen datos de calidad del aire medidos por distintas estaciones de control en Valladolid, accesibles a través del portal de Control de la Calidad del Aire de la Junta de Castilla y León. Los datos se actualizan cada día laboral, con los datos del día anterior.

Existe una fuente alternativa, disponible a través del portal de RCCAVA del ayuntamiento de Valladolid[4], sin embargo se ha optado por utilizar el portal de la Junta debido a las siguientes ventajas de esta fuente respecto a la del ayuntamiento:

- Se publican datos de las 24h anteriores (siempre que el día de actualización sea laborable).
- Se ha observado mayor fiabilidad en la disponibilidad de datos para ciertas fechas.
- El formato de salida (JSON) es más fácil de integrar con otros procesos ETL que el formato de salida ofrecido por el ayuntamiento de Valladolid (excels).
- Existe un mayor número de estaciones de medida disponibles, específicamente estaciones que no están operativas en la actualidad, así como estaciones fuera de la ciudad de Valladolid (por ejemplo estaciones de Medina del Campo o Renault).

Existen, sin embargo, ciertos inconvenientes a tener en cuenta, específicamente:

[4]https://www.valladolid.es/en/rccava/datos-red Visitado por última vez el 18 de Septiembre de 2020

- Los datos no se actualizan durante fines de semana, por lo que los datos de calidad del aire referentes a Viernes, y Sábado deben ser extraídos el Lunes. Los datos tampoco se actualizan durante días festivos.
- Las lecturas son significativamente más lentas que en el caso del ayuntamiento de Valladolid.
- No se ofrecen datos de los contaminantes de tipo BTX (Benceno, Tolueno, etc.). Sin embargo, en Valladolid estos contaminantes solo se miden en una estación (Arco de Ladrillo II).

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/ESCO/YYYY/MM/*, con *YYYY* y *MM* correspondiendo al año y mes en el que se han producido los datos
Nombre	Los datos se almacenan en ficheros csv, uno por día y por estación, con nombre *DD-idDailyObs.csv*, con *DD* correspondiendo al día en el que se han producido los datos e *id* al identificador de la estación de medida
Carga	De Martes a Viernes se extraen los datos del día anterior de la fuente y se almacenan en HDFS, los Lunes se extraen los datos de los tres días anteriores y se almacenan en HDFS. Esta extracción se realiza a las 14:00h
Actualización	Los datos se actualizan en la fuente a las 13:00 del día **laboral** siguiente a su recogida
Crecimiento	El crecimiento es lineal, aumentando alrededor de 50KB cada día

Atributos				
Índice	**Nombre**	**Tipo**	**Descripción**	**Nullable**
3.1	**no**	Float	Valor medio ($\mu g/m^3$) detectado de Óxido Nítrico durante la hora indicada	SI
3.2	**o3**	Float	Valor medio ($\mu g/m^3$) detectado de Ozono durante la hora	SI
3.3	**nombreProvincia**	String	Nombre de la provincia	NO
3.4	**pm10**	Float	Valor medio ($\mu g/m^3$) detectado de partículas de diámetro inferior a $10\mu m$ durante la hora indicada	SI
3.5	**idEstacion**	Integer	Identificador de la estación de medida	NO
3.6	**co**	Float	Valor medio (mg/m^3) detectado de Monóxido de Carbono durante la hora indicada	SI
3.7	**no2**	Float	Valor medio ($\mu g/m^3$) detectado de Dióxido Nitroso durante la hora indicada	SI
3.8	**pst**	Float	Valor medio detectado de partículas suspendidas durante la hora indicada	SI
3.9	**fecha**	String	Fecha y hora durante la que se midieron los datos	NO
3.10	**pm25**	Float	Valor medio ($\mu g/m^3$) detectado de partículas de diámetro menor que 2,5μm durante la hora indicada	SI
3.11	**codProvincia**	Integer	Identificador de la provincia	NO
3.12	**so2**	Float	Valor medio ($\mu g/m^3$) detectado de Dióxido de Azufre durante la hora indicada	SI
3.13	**nombreEstacion**	String	Nombre de la estación de medida	NO
3.14	**sh2**	Float	Valor medio detectado de Ácido Sulfhídrico durante la hora indicada	SI
3.15	**timestamp_value**	String	Timestamp correspondiente a la fecha y hora durante la que se midieron los datos	NO

Tabla 4.4: Perfil de datos en bruto - Mediciones de calidad del aire

4.1.4. Umbrales de calidad del aire

El ayuntamiento de Valladolid ofrece una tabla resumen con valores umbral para distintos contaminantes[5], unión de los umbrales establecidos por el Real Decreto 102/2011[30][6] así como la Guía de calidad del aire de la OMS [32].

Algunos de estos umbrales tienen características que no permiten ser detectados automáticamente (restricciones de superficie de la zona medida, medidas móviles, medias sub-horarias etc.). Sin embargo, aquellos que no tienen estas características, se han extraído de la tabla manualmente, generando una lista de umbrales de tres tipos: horarios, media diaria y media anual.

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/Umbrales*
Nombre	Los datos se almacenan en un único fichero csv *Umbrales.csv*
Carga	Se lleva a cabo manualmente
Actualización	No se prevé actualización
Crecimiento	No se prevé crecimiento

Atributos				
Índice	**Nombre**	**Tipo**	**Descripción**	**Nullable**
4.1	**contaminante**	String	Contaminante para el que se define el umbral	NO
4.2	**temporalidad**	String	Define la agregación de datos para la que se define el umbral, puede tener los valores *{horaria, diaria, anual}*	NO
4.3	**valor**	Float	Valor límite umbral	NO

Tabla 4.5: Perfil de datos en bruto - Umbrales

4.1.5. Mediciones meteorológicas

La Agencia Estatal de Meteorología publica una serie de medidas meteorológicas como la temperatura del aire, la velocidad del viento, la humedad relativa, etc. con una granularidad horaria para cada una de sus estaciones de medida en el territorio español. Sin embargo, a diferencia de las mediciones de calidad del aire, no existe un histórico para estas mediciones, por lo que las mediciones solo se mantienen disponibles durante las 23 horas posteriores a su realización. Por ello para esta fuente solo existen datos a partir de Noviembre de 2019.

En el caso de Valladolid, existe una estación situada en la ciudad, cuyos datos se pueden consultar de dos

[5] https://www.valladolid.es/en/rccava/normativa/limites-umbrales Visitado por última vez el 18 de Septiembre de 2020

[6] Este RD ha sido modificado por el Real Decreto 39/2017 [31], sin embargo, los valores umbral continúan siendo los del decreto de 2011.

formas, vía web[7] de manera tabular, con la opción de exportar los datos a excel o CSV, o a través de una API[8].

Mientras que la recogida de datos se comenzó a realizar automatizando la descarga a través de la web, se decidió pasar a realizarla a través de la API debido a:

- Un mayor número de variables disponibles.
- Mayor simplicidad en la lectura.
- Un inventario completo de metadatos para todas las variables disponibles.

Sin embargo tiene también algún inconveniente:

- Descarga en formato JSON, se debe convertir a un formato tabular como preprocesamiento al almacenamiento en la landing zone.
- Necesaria API Key para el acceso a los datos.

El perfil de datos ofrecido a continuación es aquel que describe los datos de la API, la descarga vía web contiene distintos nombres y un subconjunto de los atributos.

Metadatos	
Localización	Directorio HDFS */user/esther/TFM/raw/AEMET/YYYY/MM* con *YYYY* y *MM* correspondiendo al año y mes en el que se han producido los datos
Nombre	Los datos se almacenan en ficheros csv, uno por día, con nombre *DDDailyObs.csv*, con *DD* correspondiendo al día en el que se han producido los datos
Carga	Todos los días a las 23:50h se extraen los datos de dicho día de la fuente y se almacenan en HDFS
Actualización	Los datos se actualizan en la fuente a las 23:00h del día de su recogida
Crecimiento	El crecimiento es lineal, aumentando alrededor de 9KB cada día

[7] http://www.aemet.es/es/eltiempo/observacion/ultimosdatos?k=cle&l=2422&w=0&datos=det Visitado por última vez el 18 de Septiembre de 2020

[8] https://opendata.aemet.es/centrodedescargas/inicio Visitado por última vez el 18 de Septiembre de 2020

Atributos				
Índice	**Nombre**	**Tipo**	**Descripción**	**Nullable**
5.1	**vv**	Float	Velocidad (m/s) media del viento, media escalar de las muestras adquiridas cada 0,25 o 1 segundo en el periodo de 10 minutos anterior al indicado	SI
5.2	**tss20cm**	Float	Temperatura (ºC) del subsuelo a una profundidad de 20cm y correspondiente a los 10 minutos anteriores a la fecha dada	SI
5.3	**dmaxu**	Float	Dirección (grados) del viento máximo registrado en los 60 minutos anteriores a la hora y fecha indicada medido por el sensor ultrasónico	SI
5.4	**pres**	Float	Presión (hPa) instantánea al nivel en el que se encuentra instalado el barómetro a la fecha y hora indicada	SI
5.5	**tamax**	Float	Temperatura (ºC) máxima del aire, valor máximo de los 60 valores instantáneos medidos en el periodo de 60 minutos anteriores a la hora y fecha indicada	SI
5.6	**idema**	Integer	Identificador de la estación meteorológica que ha recogido los datos	NO
5.7	**lon**	Float	Longitud geográfica (grados) de la estación meteorológica	NO
5.8	**hr**	Float	Humedad relativa (%) instantánea del aire correspondiente a la hora y fecha indicada	SI
5.9	**dv**	Float	Dirección (grados) media del viento en el periodo de 10 minutos anteriores a la hora y fecha indicada	SI
5.10	**prec**	Float	Precipitación acumulada (mm, equivalente a l/m^2), medida por el pluviómetro, durante los 60 minutos anteriores a la hora y fecha indicada	SI
5.11	**tamin**	Float	Temperatura (ºC) mínima del aire, valor mínimo de los 60 valores instantáneos medidos en el periodo de 60 minutos anteriores a la hora y fecha indicada	SI
5.12	**vmax**	Float	Velocidad (m/s) máxima del viento, valor máximo del viento mantenido durante 3 segundos y registrado en los 60 minutos anteriores a la hora y fecha indicada	SI
5.13	**lat**	Float	Latitud geográfica (grados) de la estación meteorológica	NO

5.14	**vis**	Float	Visibilidad (km) promedio durante los 10 minutos anteriores a la fecha dada	SI
5.15	**dvu**	Float	Dirección (grados) media del viento, medida por el sensor ultrasónico, en el periodo de 10 minutos anteriores a la hora y fecha indicada	SI
5.16	**fint**	String	Fecha y hora final del periodo de observación (YYYY-MM-DD'T'HH:MM:SS)	NO
5.17	**alt**	Float	Altitud (m) de la estación	NO
5.18	**pres_nmar**	Float	Presión (hPa) instantánea reducida al nivel del mar a la fecha y hora indicada. Disponible para aquellas estaciones cuya altitud es igual o menor a 750 metros	SI
5.19	**stdvv**	Float	Desviación estándar (m/s) de las muestras adquiridas de velocidad del viento durante los 10 minutos anterior a la hora y fecha indicada	SI
5.20	**ta**	Float	Temperatura (ºC) instantánea del aire correspondiente a la hora y fecha indicada	SI
5.21	**stddvu**	Float	Desviación estándar (grados) de las muestras adquiridas de dirección del viento durante los 10 minutos anterior a la hora y fecha indicada medida por el sector ultrasónico	SI
5.22	**dmax**	Float	Dirección (grados) del viento máximo registrado en los 60 minutos anteriores a la hora y fecha indicada	SI
5.23	**inso**	Float	Duración (horas) de la insolación durante los 60 minutos anteriores a la hora y fecha indicada	SI
5.24	**tss5cm**	Float	Temperatura (ºC) del subsuelo a una profundidad de 5cm, correspondiente a los 10 minutos anteriores a la hora y fecha indicada	SI
5.25	**vvu**	Float	Velocidad (m/s) media del viento, media escalar de las muestras adquiridas cada 0,25 o 1 segundo en el periodo de 10 minutos anterior al indicado medida por el sensor ultrasónico	SI
5.26	**vmaxu**	Float	Velocidad (m/s) máxima del viento, valor máximo del viento mantenido durante 3 segundos y registrado en los 60 minutos anteriores a la hora y fecha indicada medida por el sensor ultrasónico	SI
5.27	**tpr**	Float	Temperatura (ºC) del punto de rocío correspondiente a la hora y fecha indicada	SI

5.28	**ubi**	String	Nombre de la estación de medición de datos	NO
5.29	**stdvvu**	Float	Desviación estándar (m/s) de las muestras adquiridas de velocidad del viento durante los 10 minutos anterior a la hora y fecha indicada medida por el sensor ultrasónico	SI
5.30	**ts**	Float	Temperatura (ºC) instantánea junto al suelo, correspondiente a los 10 minutos anteriores a la hora y fecha indicada	SI
5.31	**stddv**	Float	Desviación estándar (grados) de las muestras adquiridas de dirección del viento durante los 10 minutos anterior a la hora y fecha indicada	SI

Tabla 4.6: Perfil de datos en bruto - Mediciones meteorológicas

4.2. Modelado de Dominio

En esta sección se ofrece una visión conceptual de los datos refinados, utilizando un esquema Entidad-Relación para diseñar aquellas entidades necesarias para el proyecto así como las relaciones entre ellas. Con el objetivo de proporcionar un mayor nivel de detalle, se incluyen también diccionarios de datos, que documentan las características de cada entidad y sus atributos.

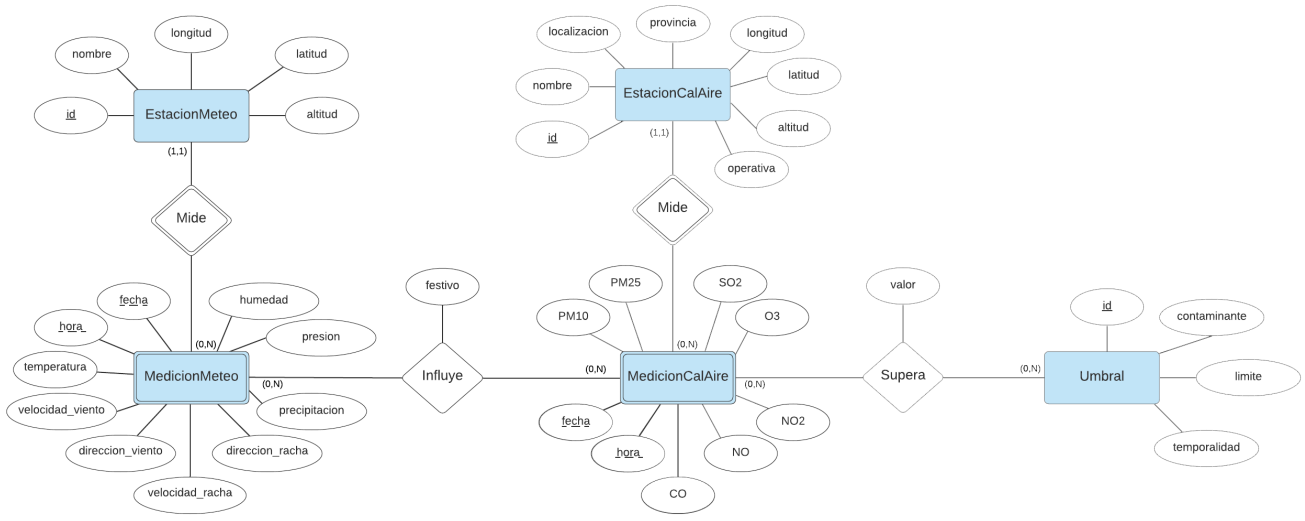


Figura 4.1: Esquema relacional del refined data

El modelo completo se puede expresar como: La entidad central, las mediciones de calidad del aire, son medidas por una estación e influidas por cero o más mediciones meteorológicas, que a su vez son medidas por una estación meteorológica. Además, las mediciones de calidad del aire pueden superar cero o más umbrales.

EstacionMeteo: Almacena información sobre las estaciones de medida de datos meteorológicos.

Campo	Tipo	Descripción	Nullable	Unique
id	INT	Atributo 5.6	NO	SI
nombre	STRING	Atributo 5.28	SI	NO
longitud	FLOAT	Atributo 5.7	SI	NO
latitud	FLOAT	Atributo 5.13	SI	NO
altitud	FLOAT	Atributo 5.17	SI	NO

Tabla 4.7: Diccionario de datos - EstacionMeteo

MedicionMeteo: Almacena variables meteorológicas medidas en cierta estación durante una fecha y hora.

Campo	Tipo	Descripción	Nullable	Unique
fecha	STRING	Fecha de recogida de datos	NO	NO
hora	INT	Hora de recogida de datos	NO	NO
idEstacion	INT	Atributo 5.6	NO	NO
temperatura	FLOAT	Atributo 5.20	SI	NO
velocidad_viento	FLOAT	Atributo 5.1	SI	NO
direccion_viento	INT	Atributo 5.9	SI	NO
velocidad_racha	FLOAT	Atributo 5.12	SI	NO
direccion_racha	INT	Atributo 5.22	SI	NO
precipitacion	FLOAT	Atributo 5.10	SI	NO
presion	FLOAT	Atributo 5.4	SI	NO
humedad	FLOAT	Atributo 5.8	SI	NO

Tabla 4.8: Diccionario de datos - MedicionMeteo

Influencia: Recoge la relación entre las mediciones meteorológicas y las mediciones de contaminantes sobre las que influyen.

Campo	Tipo	Descripción	Nullable	Unique
fecha_meteo	STRING	Fecha durante la que se han medido los datos meteorológicos	SI	NO
hora_meteo	INT	Hora durante la que se han medido los datos meteorológicos	SI	NO
idEstacion_meteo	INT	Id de la estación de medida meteorológica	SI	NO
fecha_calaire	STRING	Fecha durante la que se han medido los datos de calidad del aire	NO	NO
hora_calaire	INT	Hora durante la que se han medido los datos de calidad del aire	NO	NO
idEstacion_calaire	INT	Id de la estación de medida de calidad del aire	NO	NO
festivo	BOOLEAN	Indica si la fecha durante la que se han medido los datos de calidad del aire es festivo o no	SI	NO

Tabla 4.9: Diccionario de datos - Influencia

EstacionCalAire: Almacena información sobre las estaciones que miden los datos de calidad del aire.

Campo	Tipo	Descripción	Nullable	Unique
id	INT	Atributo 2.9	NO	SI
nombre	STRING	Atributo 2.1	SI	NO
localizacion	STRING	Atributo 2.2	SI	NO
provincia	STRING	Atributo 2.3	NO	NO
longitud	FLOAT	Longitud geográfica en la que está la estación expresada en grados decimales	SI	NO
latitud	FLOAT	Latitud geográfica en la que está la estación expresada en grados decimales	SI	NO
altitud	FLOAT	Atributo 2.6	SI	NO
operativa	BOOLEAN	Atributo 2.7	NO	NO

Tabla 4.10: Diccionario de datos - EstacionCalAire

MedicionCalAire: Almacena variables sobre contaminantes medidas en cierta estación durante una fecha y hora.

Campo	Tipo	Descripción	Nullable	Unique
fecha	STRING	Fecha durante la que se midieron los datos	NO	NO
hora	INT	Hora durante la que se midieron los datos	NO	NO
idEstacion	INT	Atributo 3.5	NO	NO
CO	FLOAT	Atributo 3.6	SI	NO
NO	FLOAT	Atributo 3.1	SI	NO
NO2	FLOAT	Atributo 3.7	SI	NO
O3	FLOAT	Atributo 3.2	SI	NO
PM10	FLOAT	Atributo 3.4	SI	NO
PM25	FLOAT	Atributo 3.10	SI	NO
SO2	FLOAT	Atributo 3.12	SI	NO

Tabla 4.11: Diccionario de datos - MedicionCalAire

SuperacionUmbral: Almacena información sobre la superación de cierto umbral, producida en una estación de medición de calidad del aire durante un periodo con inicio en una fecha y hora.

Campo	Tipo	Descripción	Nullable	Unique
idUmbral	INT	Identificador del umbral que ha sido superado	NO	NO
fecha	STRING	Primera fecha del periodo durante el que se ha superado el umbral	NO	NO
hora	INT	Primera hora del periodo durante el que se ha superado el umbral	NO	NO
idEstacion	INT	Estación de medida de contaminantes en la que se ha superado el umbral	NO	NO
valor	FLOAT	Valor que ha causado la superación del umbral	NO	NO

Tabla 4.12: Diccionario de datos - SuperacionUmbral

Umbral: Almacena datos sobre los límites definidos para distintos contaminantes y agregaciones temporales.

Campo	Tipo	Descripción	Nullable	Unique
id	STRING	Identificador del umbral	NO	SI
contaminante	STRING	Atributo 4.1	NO	NO
temporalidad	STRING	Atributo 4.2	NO	NO
limite	FLOAT	Atributo 4.3	NO	NO

Tabla 4.13: Diccionario de datos - Umbral

4.3. Diseño del Dataflow

En el diseño del dataflow se muestra el esquema lógico del refined data (Figura 4.2), aquel que expresa el resultado final que tendrá la transformación incluyendo la **desnormalización de los datos**.

La desnormalización se basa en almacenar en una misma tabla datos que ya están representados en otra tabla, y a los que tradicionalmente se accedería a través de una relación de clave foránea. Mientras que en los sistemas de bases de datos relacionales uno de los requisitos imprescindibles es la normalización de los datos, el Data Lake es un sistema desnormalizado.

En nuestro caso esta desnormalización tiene su motivación en dos factores: primero, la tecnología HDFS/Hadoop no está optimizada para estructuras normalizadas; y segundo, en este trabajo se utilizan solamente las lecturas de una estación de meteorología, y se va a asumir que tienen influencia en la calidad del aire sólo aquellas producidas durante el mismo periodo de tiempo, por tanto es innecesaria una separación entre las tablas MedicionCalAire y MedicionMeteo.

En la desnormalización propuesta se incluye dentro de la tabla MedicionCalAire también las mediciones meteorológicas obtenidas en ese mismo instante de tiempo (fecha y hora), así como si dicha fecha es o no festivo. De esta manera se evita que el sistema tenga que realizar costosos JOINs entre las tablas para obtener los datos de calidad del aire y meteorología relacionados. Esta desnormalización está reflejada a partir de ahora en el esquema y mapa lógico de datos.

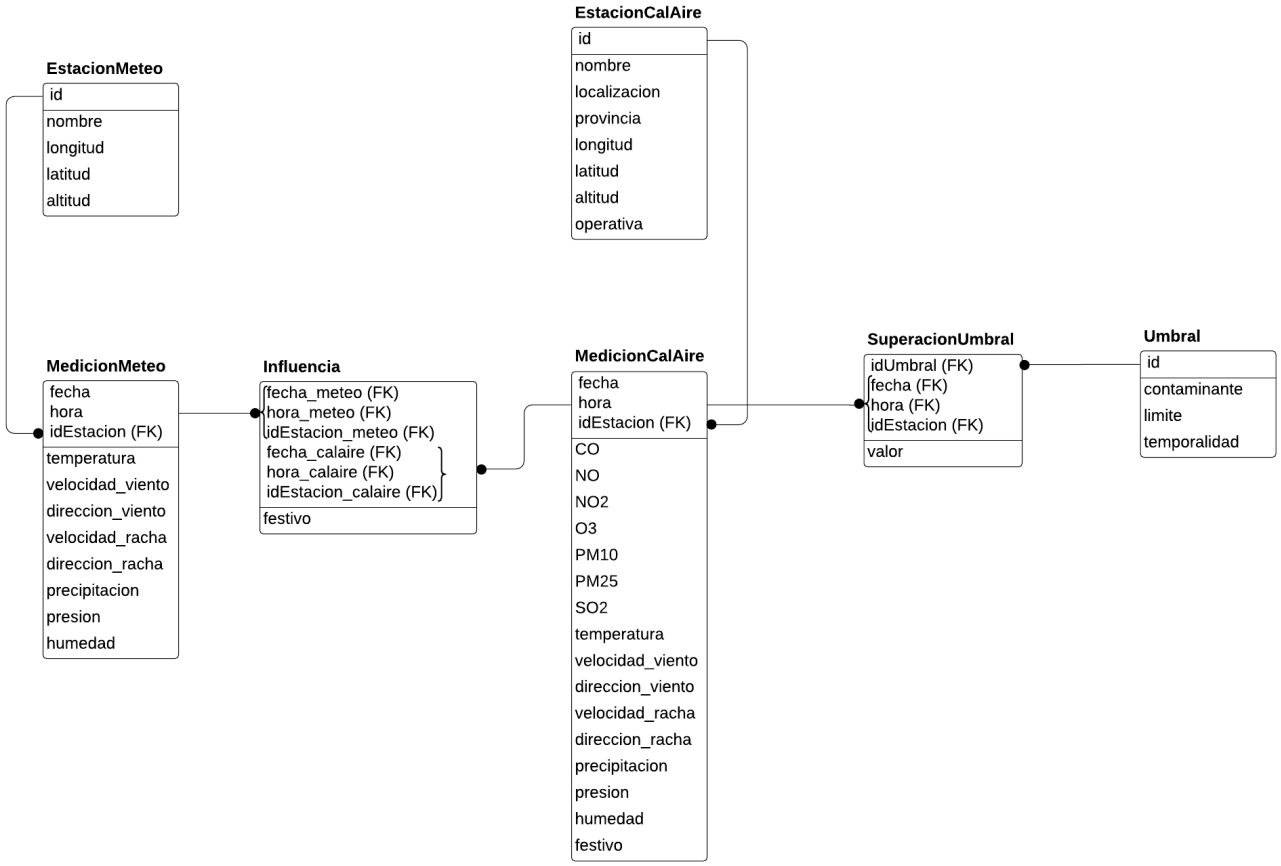


Figura 4.2: Esquema lógico del refined data

Pasamos tras esto a hablar del workflow de transformación de los datos, mostrado en la Figura 4.3, así como la descripción de las operaciones realizadas en el mismo. Como podemos observar, los datos que se almacenan en el Data Lake son **todos aquellos resultantes de cada transformación**, y se distribuye en varias zonas. Esto permite organizar los datos de manera lógica para usuarios, así como establecer roles adecuados para cada zona (por ejemplo, la zona donde estarán los datos refinados, conocida como **Data Hub** puede que sólo sea accesible a analistas, mientras que las capas anteriores deben ser accesibles a ingenieros o desarrolladores que puedan trabajar en ellas). Se describen las capas a continuación:

- **Raw**: También conocida como «landing» o capa de aterrizaje, almacena los datos en bruto tras ser ingestados en el sistema. Estos datos suelen estar particionados por ejemplo por el día de recogida, simplemente para evitar directorios con un gran número de ficheros.
- **L0 - Alignment**: Esta es la primera capa de trabajo, la que corresponde con la alineación de los tipos,

nombres y formatos respecto al modelo propuesto en los datos refinados. Un ejemplo de una transformación aplicada en esta capa sería: «En los datos refinados, la fecha y hora está modelada como dos atributos distintos, mientras que en los datos en bruto está representada en una sola variable. Dividimos dicha variable en su parte fecha y su parte hora.»

- **L1 - Enrichment**: La segunda etapa, realiza transformaciones y uniones entre datos para crear nuevas variables o tablas de utilidad, enriqueciendo así los datos. Un ejemplo sería: «Los datos refinados incluyen un inventario de estaciones meteorológicas. Utilizamos las mediciones meteorológicas para encontrar todas las estaciones que han medido alguna vez datos, y extraemos sus características».
- **L2 - Integration**: Por último, se integran aquellas tablas relacionadas en el modelo de datos refinado. Por ejemplo: «Las mediciones de calidad del aire se relacionan con aquellos umbrales que superan, calculamos esta relación y creamos una tabla que la representa».
- **DataHub**: También se conoce como «gold zone» o capa de oro, ya que contiene los datos refinados tras su alineamiento, enriquecimiento e integración.

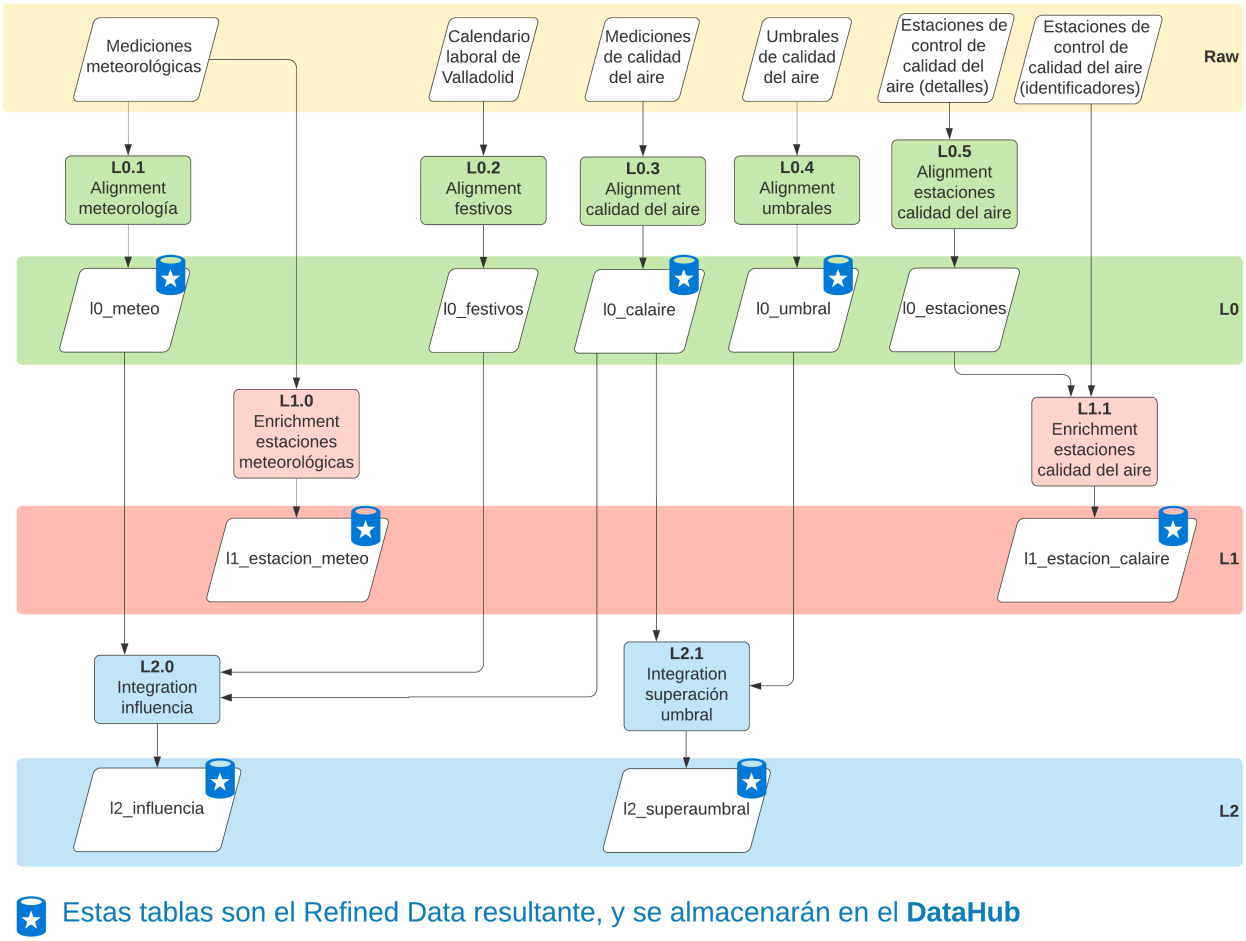


Figura 4.3: Workflow de la transformación de datos desde el raw data hasta el refined data

L0 - Alignment En esta etapa se transforman los datos en bruto para que satisfagan el modelo de datos propuesto para el refined data.

L0.1 Alignment meteorología A partir de la tabla de datos brutos MEDICIONES METEOROLÓGICAS se genera la tabla **l0_meteo**. Se descartan campos que no se utilizan, se cambian nombres y tipos, y se pasa la fecha a un formato estándar de Hive.

Campo	Acción	Campo resultado	Observaciones
idema	Cambio de nombre	idEstacion	
fint	Cambio de nombre y extracción de la fecha	fecha	El campo pasa del formato yyyy/MM/dd HH:mm:ss a yyyy-MM-dd
fint	Cambio de nombre, extracción de la hora y cambio de tipo	hora	El campo pasa del formato yyyy/MM/dd HH:mm:ss a H, y cambia de tipo STRING a INT
ta	Cambio de nombre	temperatura	
vv	Cambio de nombre	velocidad_viento	
dv	Cambio de nombre y tipo	direccion_viento	El campo pasa de tipo FLOAT a INT
vmax	Cambio de nombre	velocidad_racha	
dmax	Cambio de nombre y tipo	direccion_racha	El campo pasa de tipo FLOAT a INT
prec	Cambio de nombre	precipitacion	
pres	Cambio de nombre	presion	
hr	Cambio de nombre	humedad	
tss20cm	Se descarta		
dmaxu	Se descarta		
tamax	Se descarta		
lon	Se descarta		
tamin	Se descarta		
lat	Se descarta		
dvu	Se descarta		
alt	Se descarta		
pres_nmar	Se descarta		
stdvv	Se descarta		
stddvu	Se descarta		
inso	Se descarta		
tss5cm	Se descarta		
vvu	Se descarta		
vmaxu	Se descarta		
tpr	Se descarta		
pacutp	Se descarta		
ubi	Se descarta		
stdvvu	Se descarta		
ts	Se descarta		
stddv	Se descarta		

Tabla 4.14: Alignment meteorología

L0.2 Alignment festivos A partir de la tabla de datos brutos Calendario laboral de Valladolid se genera la tabla **l0_festivos**. Se convierten las fechas de festivos al formato de fecha estándar de Hive.

Campo	Acción	Campo resultado	Observaciones
fecha	Cambio de formato	fecha	El campo pasa del formato dd-MM-yyyy a yyyy-MM-dd

Tabla 4.15: Alignment festivos

L0.3 Alignment calidad del aire A partir de la tabla de datos brutos Mediciones de calidad del aire se genera la tabla **l0_calaire**. Se descartan campos que no se utilizan o son siempre NULL, y se cambia la fecha al formato estándar de Hive.

Campo	Acción	Campo resultado	Observaciones
idEstacion	=	idEstacion	
fecha	Extracción de la fecha	fecha	El campo pasa del formato dd/MM/yyyy HH:mm:ss a yyyy-MM-dd
fecha	Extracción de la hora, cambio de nombre y cambio de tipo	hora	El campo pasa del formato dd/MM/yyyy HH:mm:ss a H, y cambia de tipo STRING a INT
co	=	**co**	
no	=	**no**	
no2	=	**no2**	
o3	=	**o3**	
pm10	=	**pm10**	
pm25	=	**pm25**	
so2	=	**so2**	
nombreProvincia	Se descarta		
pst	Se descarta		Es siempre NULL para las estaciones y temporalidad seleccionada
codProvincia	Se descarta		
nombreEstacion	Se descarta		
sh2	Se descarta		Es siempre NULL para las estaciones y temporalidad seleccionada
timestamp_value	Se descarta		

Tabla 4.16: Alignment calidad del aire

L0.4 Aligment umbrales A partir de la tabla de datos brutos UMBRALES DE CALIDAD DEL AIRE se genera la tabla **l0_umbral**. Se renombran las variables y se asigna un identificador único autogenerado a cada umbral:

Campo	Acción	Campo resultado	Observaciones
	Autogenerado mediante la función java.util.UUID	id	
contaminante	=	contaminante	
temporalidad	=	temporalidad	
valor	Cambio de nombre	limite	

Tabla 4.17: Alignment calidad del aire

L0.5 Aligment estaciones calidad del aire A partir de la tabla de datos brutos ESTACIONES DE CONTROL DE CALIDAD DEL AIRE (DETALLES) se genera la tabla **l0_estaciones**. Se filtran los datos a la provincia de Valladolid, y se cambian nombres y tipos:

Campo	Acción	Campo resultado	Observaciones
estacion	Cambio de nombre	nombre	
localizacion	=	localizacion	
provincia	Filtrado a la provincia de Valladolid	provincia	estaciones.provincia = 'VALLADOLID'
longitud	Conversión del formato Gº M' S" H a grados decimales, cambio de tipo	longitud	El campo pasa de tipo STRING a FLOAT
latitud	Conversión del formato Gº M' S" H a grados decimales, cambio de tipo	latitud	El campo pasa de tipo STRING a FLOAT
altitud	Supresión del caracter «m» final, cambio de tipo	altitud	El campo pasa de tipo STRING a FLOAT
operativa	Transformación del valor «No» a False y «Si» a True, cambio de tipo	operativa	El campo pasa de tipo STRING a BOOLEAN

Tabla 4.18: Alignment estaciones calidad del aire

L1 - Enrichment En esta etapa se utilizan los datos ya alineados para obtener nueva información que ayude a enriquecerlos.

L1.0 Enrichment estaciones meteorológicas A partir de la tabla de datos brutos MEDICIONES METEOROLÓGICAS se genera la tabla **l1_estacion_meteo**. Se separan de los datos de mediciones las características de las estaciones que las recogieron, obteniendo los conjuntos distintos de estos 5 elementos:

Campo	Acción	Campo resultado	Observaciones
idema	Capturado de *Mediciones meteorológicas*, cambio de nombre	id	
ubi	Capturado de *Mediciones meteorológicas*, cambio de nombre	nombre	
lon	Capturado de *Mediciones meteorológicas*, cambio de nombre	longitud	
lat	Capturado de *Mediciones meteorológicas*, cambio de nombre	latitud	
alt	Capturado de *Mediciones meteorológicas*, cambio de nombre	altitud	

Tabla 4.19: Enrichment estaciones meteorológicas

L1.1 Enrichment estaciones calidad del aire A partir de la tabla de datos brutos ESTACIONES DE CONTROL DE CALIDAD DEL AIRE (IDENTIFICADORES) y la tabla alineada *l0_ estaciones* se genera la tabla **l1_estacion_calaire**. Se añade el identificador de cada estación combinando ambas tablas por el nombre de la estación:

Campo	Acción	Campo resultado	Observaciones
id	Capturado de *Estaciones de control de calidad del aire (identificadores)*	id	l0_estaciones. nombre = estaciones_ids. estacion
nombre	Capturado de l0_estaciones	nombre	
localizacion	Capturado de l0_estaciones	localizacion	
provincia	Capturado de l0_estaciones	provincia	
longitud	Capturado de l0_estaciones	longitud	
latitud	Capturado de l0_estaciones	latitud	
altitud	Capturado de l0_estaciones	altitud	
operativa	Capturado de l0_estaciones	operativa	

Tabla 4.20: Enrichment estaciones calidad del aire

L2 - Integration En esta etapa se integran los flujos de datos alineados y enriquecidos.

L2.0 Integration influencia A partir de las tablas de datos alineadas *l0_ meteo*, *l0_festivos* y *l0_ calaire*, se genera la tabla **l2_influencia**. Se establece la relación de influencia entre los datos de meteorología y los datos de calidad del aire ocurridos en la misma fecha y hora, así como la influencia de que dicha fecha sea festivo:

Campo	Acción	Campo resultado	Observaciones
fecha	Capturado de l0_meteo	fecha_meteo	l0_meteo.fecha = l0_calaire.fecha
hora	Capturado de l0_meteo	hora_meteo	l0_meteo.hora = l0_calaire.hora
idEstacion	Capturado de l0_meteo	idEstacion_meteo	
fecha	Capturado de l0_calaire	fecha_calaire	l0_meteo.fecha = l0_calaire.fecha
hora	Capturado de l0_calaire	hora_calaire	l0_meteo.hora = l0_calaire.hora
idEstacion	Capturado de l0_calaire	idEstacion_calaire	
fecha	Capturado de l0_festivos, se detecta si la fecha de calidad del aire está entre las fechas de l0_festivos	festivo	l0_calaire.fecha IN l0_festivos.fecha

Tabla 4.21: Integration influencia

L2.0 Integration superación umbral A partir de las tablas de datos alineadas *l0_umbral* y *l0_calaire*, se genera la tabla **l2_superaumbral**. Por cada umbral, se explora la tabla de calidad del aire, detectando si en la temporalidad indicada, para un contaminante determinado se supera el umbral:

Campo	Acción	Campo resultado	Observaciones
id	Capturado de l0_umbral	idUmbral	
fecha	Capturado de l0_calaire	fecha	Primera fecha del periodo durante el que se supera el umbral
hora	Capturado de l0_calaire	hora	Primera hora del periodo durante el que se supera el umbral
idEstacion	Capturado de l0_calaire	idEstacion	
	Capturado de l0_calaire	valor	Será el valor con el que se ha superado el umbral, dependerá tanto del contaminante definido en el umbral como de la temporalidad

Tabla 4.22: Integration superación umbral

Refined data En esta etapa se realiza la desnormalización y almacenado de las tablas en el DataHub para su posterior explotación.

l1_estacion_meteo, l0_meteo, l2_influencia, l1_estacion_calaire, l2_superaumbral y l0_umbral se almacenan

directamente en el DataHub como **DH_EstacionMeteo, DH_MedicionMeteo, DH_Influencia, DH_EstacionCalaire, DH_SuperaUmbral** y **DH_Umbral**, conservando todos sus campos.

A partir de la tabla *l0_calaire, l0_meteo* y *l2_influencia* se genera la tabla **DH_MedicionCalAire:**

Campo	Acción	Campo resultado	Observaciones
idEstacicon	Capturado de l0_calaire	idEstacion	
fecha	Capturado de l0_calaire	fecha	
hora	Capturado de l0_calaire	hora	
co	Capturado de l0_calaire	co	
no	Capturado de l0_calaire	valor	
no2	Capturado de l0_calaire	no2	
o3	Capturado de l0_calaire	o3	
pm10	Capturado de l0_calaire	pm10	
pm25	Capturado de l0_calaire	pm25	
so2	Capturado de l0_calaire	so2	
temperatura	Capturado de l0_meteo	temperatura	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
temperatura	Capturado de l0_meteo	temperatura	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
velocidad_viento	Capturado de l0_meteo	velocidad_viento	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
direccion_viento	Capturado de l0_meteo	direccion_viento	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
velocidad_racha	Capturado de l0_meteo	velocidad_racha	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo

direccion_racha	Capturado de l0_meteo	direccion_racha	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
precipitacion	Capturado de l0_meteo	precipitacion	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
presion	Capturado de l0_meteo	presion	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
humedad	Capturado de l0_meteo	humedad	Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia, y un LEFT OUTER JOIN con l0_meteo
festivo	Capturado de l2_influencia		Capturado mediante un LEFT JOIN de l0_calaire y l2_influencia

Tabla 4.23: Desnormalización de los datos

Capítulo 5

ETL

En este capítulo se desarrolla la implementación del proceso ETL diseñado en el capítulo 4, teniendo en cuenta las herramientas utilizadas, indicadas en el capítulo 3. Así, contamos con una sección para la etapa de Extracción, donde se explica el proceso de ingesta y almacenamiento de los datos desde su fuente; Transformación, donde exploramos cada una de las etapas de procesamiento necesarias para alinear, enriquecer e integrar los datos brutos; y por último la Carga, momento en el que los datos enriquecidos pasan al entorno de explotación.

5.1. Extracción

El proceso de extracción es aquel por el cual se explotan una o más fuentes de datos, abarcando desde su descarga del sistema de origen a su almacenamiento en un formato preparado para ser transformado.

5.1.1. Ingesta

Como hemos visto en la sección 4.1, para este proyecto se han usado cinco fuentes de datos, con tres métodos de ingesta principales, que explicamos a continuación ordenados de menos a más automatizado:

1. Inputar manualmente los datos desde la fuente a un fichero csv: Se utiliza solamente en aquellas fuentes de pequeño tamaño y cuya automatización es complicada. Específicamente se extraen de esta manera el **Calendario laboral de Valladolid**, los **Umbrales de calidad del aire** y los **Identificadores de las estaciones de control de calidad del aire**. Estos últimos requieren explorar el código fuente del portal de Control de Calidad del Aire de la Junta de Castilla y León, ya que no aparecen reflejados de manera descubierta en ningún lugar de la página. Los ficheros resultantes se suben a HDFS mediante la interfaz de Hue.

2. Descarga manual de ficheros csv: Sólo se utiliza este método para obtener el inventario de **Estaciones de control de calidad del aire.** Este método podría ser automatizado aunque, debido a que no se prevé actualización, se ha optado por no llevarlo a cabo para este proyecto. El fichero resultante se sube a HDFS mediante la interfaz de Hue.

3. Obtención automatizada y programada de datos: Este método, completamente automatizado y por tanto programable, se lleva a cabo para la descarga diaria de las mediciones horarias de **Calidad del aire** y **Meteorología**. A continuación se explora más en profundidad la implementación llevada a cabo para la ingesta de estas dos fuentes.

Mediciones de calidad del aire Como ya se ha indicado anteriormente, la fuente original de estos datos es el portal de Control de Calidad del Aire de la Junta de Castilla y León. Esta página web no ofrece API ni, aparentemente, una manera de automatizar la extracción de los datos, por lo que se comenzó explorando las llamadas que realizaba el portal al hacer una consulta de datos históricos. A continuación se muestra el contenido de la pestaña de *Network* en las herramientas de desarrollador de Google Chrome cuando se realiza una consulta de datos horarios a la estación Arco de Ladrillo entre el 1 y el 4 de Agosto de 2020:

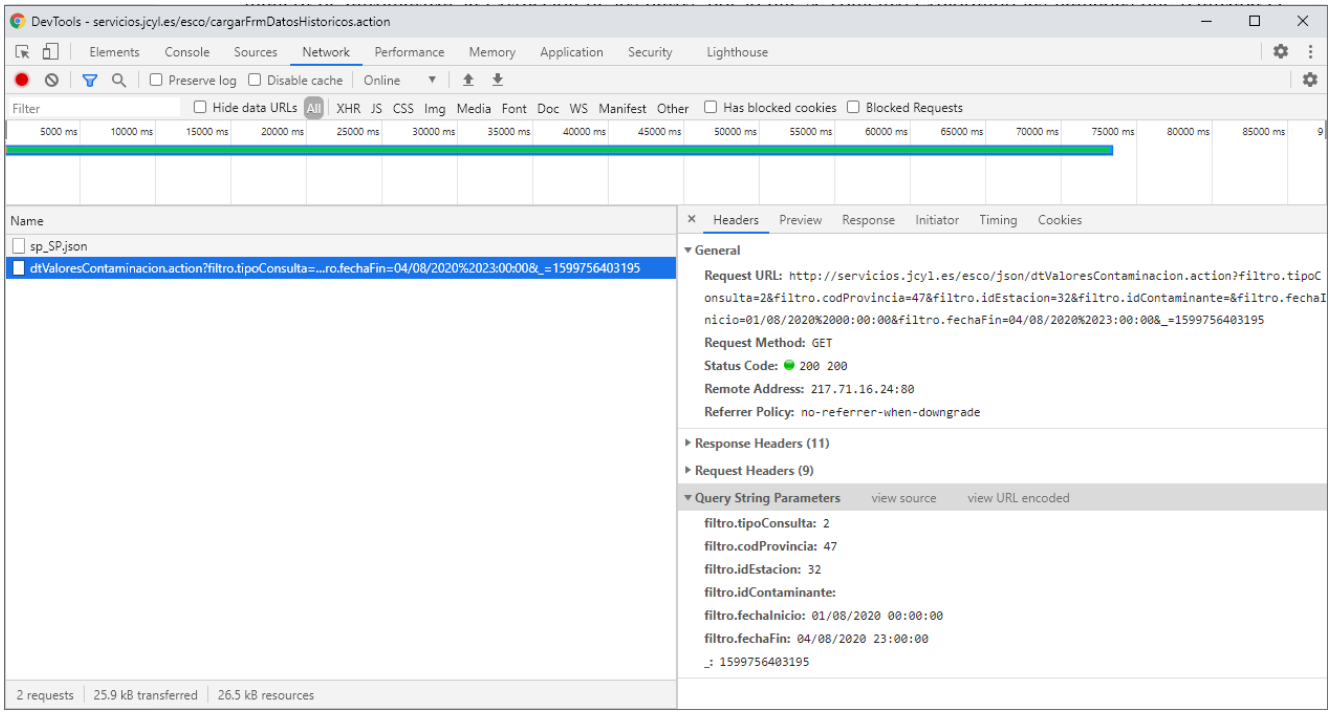


Figura 5.1: Resultado de inspeccionar las llamadas del portal de Calidad del Aire

Como se puede observar, la URL `http://servicios.jcyl.es/esco/json/dtValoresContaminacion.action` acepta peticiones tipo GET, con los parámetros siguientes:

- *filtro.tipoConsulta*: Sirve para denotar el nivel de agregación de la consulta, en el caso de la consulta horaria es un 2.
- *filtro.codProvincia:* Código de la provincia deseada, Valladolid es el 47.
- *filtro.idEstacion:* Código de la estación de la que deseamos los datos.
- *filtro.fechaInicio*: Fecha de inicio del rango de datos deseado en formato DD/MM/YYYY HH:mm:ss
- *filtro.fechaFin:* Fecha fin del rango en el mismo formato.

La respuesta a esta petición es un fichero JSON:


```
{
"data": [
        {
        "co": 0.3,
      "codProvincia": "47",
      "fecha": "01/08/2020␣00:00:00",
      "idEstacion": 32,
      "no": 3.0,
      "no2": 16.0,
      "nombreEstacion": "Arco␣de␣ladrillo␣II",
      "nombreProvincia": "Valladolid",
      "o3": null,
      "pm10": 23.0,
      "pm25": 19.0,
      "pst": null,
      "sh2": null,
      "so2": null,
      "timestamp": 1596232800000
    },
...
```


Esto nos permite elaborar un programa en Java que realiza consultas a esta dirección, lee los resultados, los convierte a un fichero CSV, y los almacena directamente en HDFS.

Mediciones meteorológicas A diferencia de la fuente anterior, la Agencia Española de Meteorología (AEMET) dispone de una API para acceder a muchos de sus servicios, entre ellos los últimos datos de observación meteorológica elaborados. Para acceder a la API es necesario disponer de una clave, que se emite de manera gratuita.

Para extraer los datos de la API se ha creado un programa en Java que hace una llamada a la URL `https://opendata.aemet.es/opendata/api/observacion/convencional/datos/estacion/2422/`, cuya respuesta es un fichero JSON con una URL desde la cual se pueden descargar otro fichero JSON con los datos:

```
[
  {
    "idema": "2422",
    "lon": -4.75441,
    "fint": "2020-09-09T18:00:00",
    "prec": 0.0,
    "alt": 735.0,
    "vmax": 4.7,
    "vv": 1.1,
    "dv": 42.0,
    "lat": 41.64083,
    "dmax": 83.0,
    "ubi": "VALLADOLID",
    "pres": 935.4,
    "hr": 24.0,
    "stdvv": 0.6,
    "ts": 26.1,
    "pres_nmar": 1016.8,
    "tamin": 26.2,
    "ta": 26.2,
    "tamax": 27.1,
    "tpr": 3.9,
    "vis": 20.0,
    "stddv": 54.0,
    "inso": 2.8,
    "tss5cm": 30.9,
    "vmaxu": 4.7,
    "dvu": 37.0,
    "pacutp": 0.0,
    "vvu": 0.9,
    "stdvvu": 0.9,
    "stddvu": 40.0,
    "dmaxu": 64.0,
    "tss20cm": 28.8
  },
...
```


El fichero es transformado al formato CSV y almacenado en HDFS.

5.1.2. Almacenamiento

Los datos del DataLake se almacenan en HDFS, lo que permite que todo el entorno Hadoop pueda acceder a ellos. Se ha seguido un esquema de carpetas que refleja la modularidad del proceso ETL, donde podemos encontrar una carpeta raíz en la que se encuentran cinco directorios:

- **raw**: Esta carpeta contiene los datos en bruto en formato CSV. Los directorios que almacenan las medidas de calidad del aire (ESCO) se encuentra particionado por año, mes y día, lo cual se muestra en la estructura de carpetas. Las últimas carpetas contienen un fichero por cada estación de recogida. A su vez las mediciones meteorológicas (AEMET) están particionadas por año y mes, con la última carpeta conteniendo un fichero por cada día con datos.
- **L0**: En este directorio se encuentran aquellos datos resultantes del primer proceso de transformación, el *Alignment*. La salida de cada etapa mencionada en la sección 4.3 genera una carpeta.
- **L1**: Almacenan los resultados de la etapa de *Enrichment*, descrita en la sección 4.3.
- **L2**: Aquí se depositan los resultados de la fase de *Integration* tal como se describe en la sección 4.3.
- **L3**: Este directorio contiene una pequeña tabla temporal que almacena los datos refinados dh_medicioncalaire y dh_superaumbral de las últimas 24 horas procesadas. Se utilizan para exportar al entorno de explotación sólo los datos que no se hayan exportado ya.

Debido a que los datos refinados se almacenan en tablas internas de Hive, los encontramos bajo la carpeta /user/hive/warehouse/tfm.db.

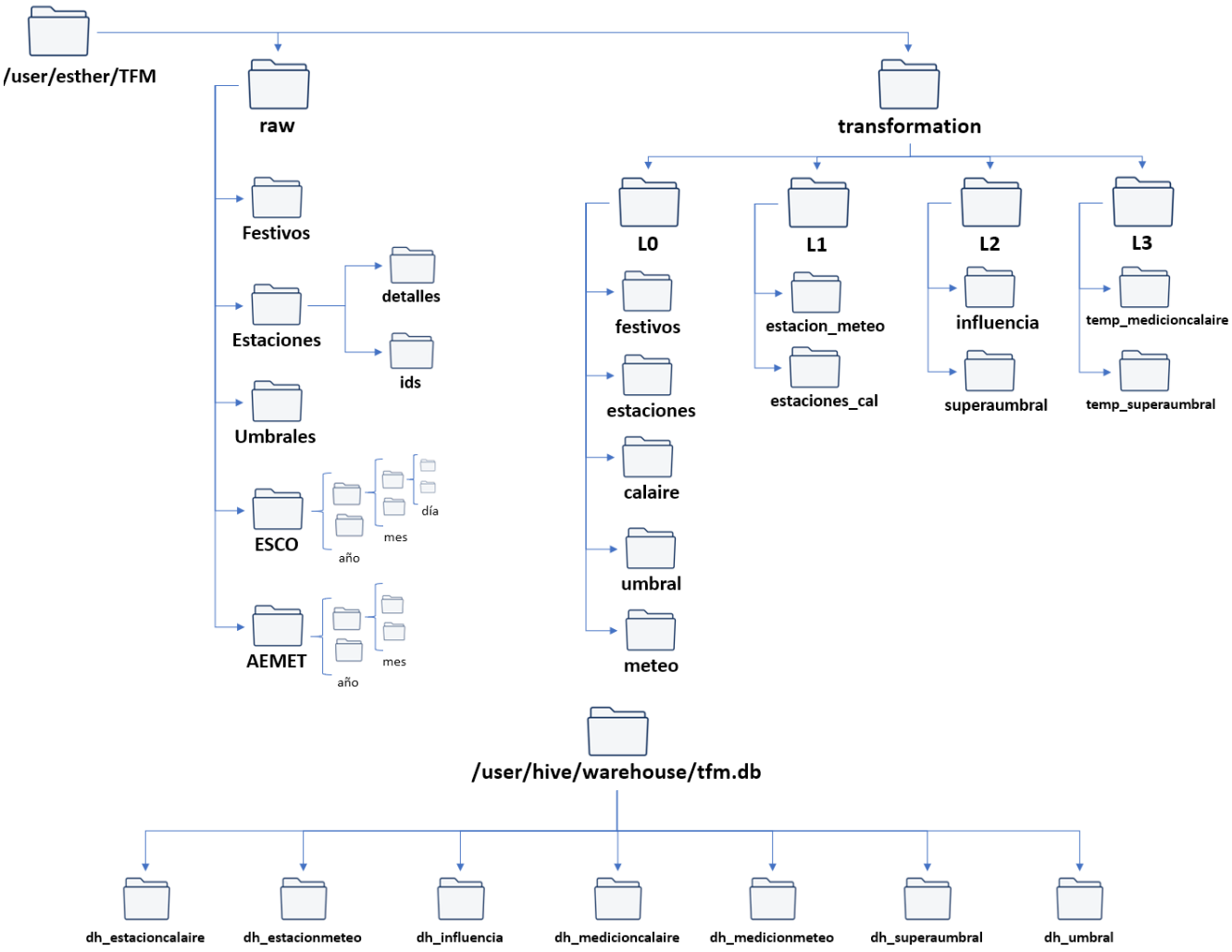


Figura 5.2: Estructura de carpetas en HDFS

5.2. Transformación

En la fase de transformación se llevan a cabo todo aquel procesamiento necesario para convertir los datos en bruto definidos en 4.1 en datos refinados modelados en 4.2 mediante la implementación del Dataflow diseñado en 4.3. Previo a esta etapa, se deben crear manualmente los directorios y tablas que contendrán los datos intermedios de cada etapa. El código usado para esta labor se encuentra en el Apéndice A.

La mayoría de las transformaciones han sido automatizadas mediante workflows y programadas para su ejecución todos los días mediante coordinadores, ambos componentes partes de *Oozie*, sin embargo, existen algunas

transformaciones que sólo requieren ser ejecutadas una vez debido a que la fuente no se va a actualizar, específicamente:

- **L0.2 Alignment festivos**:

Listado de código 5.1: Código necesario para llevar a cabo L0.2

```
INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L0/festivos' ROW
    FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
    AS TEXTFILE
SELECT to_date(from_unixtime(unix_timestamp(festivos.fecha, "dd-MM-yyyy")))
FROM festivos;
```

- **L0.4 Alignment umbrales:**

Listado de código 5.2: Código necesario para llevar a cabo L0.4

```
INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L0/umbral' ROW
    FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
    AS TEXTFILE
SELECT reflect("java.util.UUID", "randomUUID") as id, contaminante,
    temporalidad, valor as limite
FROM umbrales;
```

- **L0.5 Alignment estaciones**:

Listado de código 5.3: Código necesario para llevar a cabo L0.5

```
CREATE VIEW  IF NOT EXISTS calculo_coordenadas AS
  SELECT DISTINCT
    longitud,
    CAST(REGEXP_EXTRACT(longitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,1) AS INT) lon_deg,
    CAST(REGEXP_EXTRACT(longitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,2) AS INT) lon_min,
    CAST(REGEXP_EXTRACT(longitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,3) AS INT) lon_seg,
    REGEXP_EXTRACT(longitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])',4)
       lon_hem,
    latitud,
    CAST(REGEXP_EXTRACT(latitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,1) AS INT) lat_deg,
    CAST(REGEXP_EXTRACT(latitud, '([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,2) AS INT) lat_min,
```

```
    CAST(REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,3) AS INT) lat_seg,
    REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])',4)
       lat_hem
  FROM estaciones;

INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L0/estaciones' ROW
    FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
   AS TEXTFILE
SELECT e.estacion, e.localizacion, e.provincia,
  IF(cc.lon_hem='W',-(cc.lon_deg+(cc.lon_min/60)+(cc.lon_seg/3600)),cc.lon_deg
     +(cc.lon_min/60)+(cc.lon_seg/3600)) as longitud,
  IF(cc.lat_hem='S',-(cc.lat_deg+(cc.lat_min/60)+(cc.lat_seg/3600)),cc.lat_deg
     +(cc.lat_min/60)+(cc.lat_seg/3600)) as latitud,
  CAST(REGEXP_EXTRACT(e.altitud,'([0-9]*)(␣)*m',1) AS FLOAT) as altitud,
  IF(operativa='Si', TRUE, FALSE) as operativa
FROM estaciones e LEFT JOIN calculo_coordenadas cc ON cc.longitud=e.longitud
   AND cc.latitud=e.latitud
WHERE e.provincia='VALLADOLID';
```

- **L0.5 Alignment estaciones**:

Listado de código 5.4: Código necesario para llevar a cabo L0.5

```
CREATE VIEW  IF NOT EXISTS calculo_coordenadas AS
  SELECT DISTINCT
    longitud,
    CAST(REGEXP_EXTRACT(longitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,1) AS INT) lon_deg,
    CAST(REGEXP_EXTRACT(longitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,2) AS INT) lon_min,
    CAST(REGEXP_EXTRACT(longitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])'
        ,3) AS INT) lon_seg,
    REGEXP_EXTRACT(longitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([EW])',4)
       lon_hem,
    latitud,
    CAST(REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,1) AS INT) lat_deg,
    CAST(REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,2) AS INT) lat_min,
    CAST(REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])'
        ,3) AS INT) lat_seg,
```

```
    REGEXP_EXTRACT(latitud,'([0-9]{2}).*([0-9]{2}).*([0-9]{2}).*([SN])',4)
      lat_hem
  FROM estaciones;

INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L0/estaciones' ROW
    FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
   AS TEXTFILE
SELECT e.estacion, e.localizacion, e.provincia,
  IF(cc.lon_hem='W',-(cc.lon_deg+(cc.lon_min/60)+(cc.lon_seg/3600)),cc.lon_deg
     +(cc.lon_min/60)+(cc.lon_seg/3600)) as longitud,
  IF(cc.lat_hem='S',-(cc.lat_deg+(cc.lat_min/60)+(cc.lat_seg/3600)),cc.lat_deg
     +(cc.lat_min/60)+(cc.lat_seg/3600)) as latitud,
  CAST(REGEXP_EXTRACT(e.altitud,'([0-9]*)( )*m',1) AS FLOAT) as altitud,
  IF(operativa='Si', TRUE, FALSE) as operativa
FROM estaciones e LEFT JOIN calculo_coordenadas cc ON cc.longitud=e.longitud
   AND cc.latitud=e.latitud
WHERE e.provincia='VALLADOLID';
```

- **L1.1 Enrichment estaciones calidad del aire:**

Listado de código 5.5: Código necesario para llevar a cabo L1.1

```
INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L1/estacion_cal'
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
   STORED AS TEXTFILE
SELECT eid.id, e.nombre, e.localizacion, e.provincia, e.longitud, e.latitud, e
   .altitud, e.operativa
FROM l0_estaciones e LEFT JOIN estaciones_ids eid ON(e.nombre=eid.estacion);
```

- **Almacenamiento en el DataHub de Umbral y EstacionCalaire:**

Listado de código 5.6: Código necesario para llevar a cabo el almacenamiento en el DataHub de Umbral y EstacionCalaire

```
INSERT OVERWRITE TABLE DH_Umbral SELECT * FROM l0_umbral;
INSERT OVERWRITE TABLE DH_EstacionCalAire SELECT * FROM l1_estacion_cal;
```

Comenzamos comentando los coordinadores. Este componente de *Oozie* permite a los usuarios ejecutar workflows de manera recurrente, de manera similar a como funciona el programa *cron* en sistemas Unix [33]. Hue permite crear estos coordinadores mediante una interfaz, aunque como el resto de componentes de *Oozie*, se basa en ficheros XML que describen el workflow a ejecutar, la recurrencia, las fechas de inicio y final, y las propiedades a introducir al workflow.

Para este proyecto se han diseñado tres coordinadores:

1. El primero extrae y procesa los datos meteorológicos, ejecutando el workflow «Daily AEMET Download». Este workflow se ejecutará todos los días a las 23:50 horas, ya que los datos se actualizan en la fuente alrededor de dicha hora. El workflow acepta tres parámetros, ${YEAR}, ${MONTH} y ${DAY}, referentes al año, mes y día de datos que se descargan respectivamente. El coordinador permite acceder al momento en el que se ejecuta el trabajo mediante la variable *coord:nominalTime()*, que se formatea para obtener cada componente de la fecha necesario.

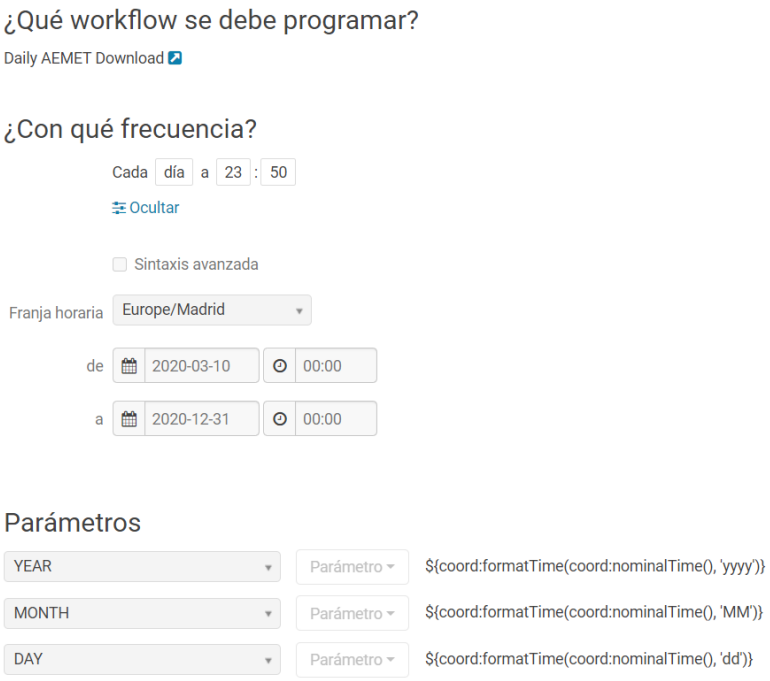


Figura 5.3: Coordinador para la extracción y procesado de los datos meteorológicos

2. El segundo extrae y procesa los datos de contaminantes, calcula la superación de umbrales, integra los datos y exporta el refined data al entorno de explotación. Este trabajo se ejecuta todos los días de Lunes a Viernes a las 13 horas, ya que es cuando se actualiza la fuente con los datos del día anterior.

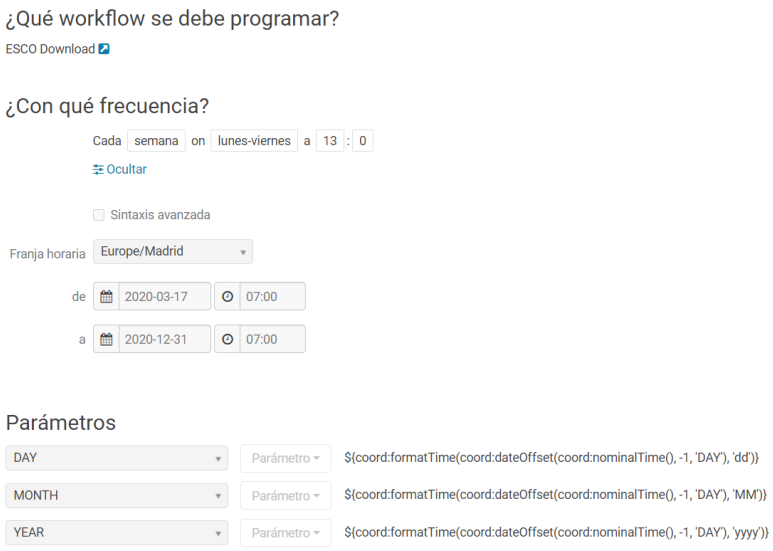


Figura 5.4: Coordinador para la extracción y procesado de los datos de ESCO entre semana

3. Debido a que la fuente de datos de contaminantes no se actualizan durante el fin de semana, el Lunes se ejecuta un workflow especial que ejecuta el workflow «ESCO Download» para descargar los datos del Viernes y Sábado:

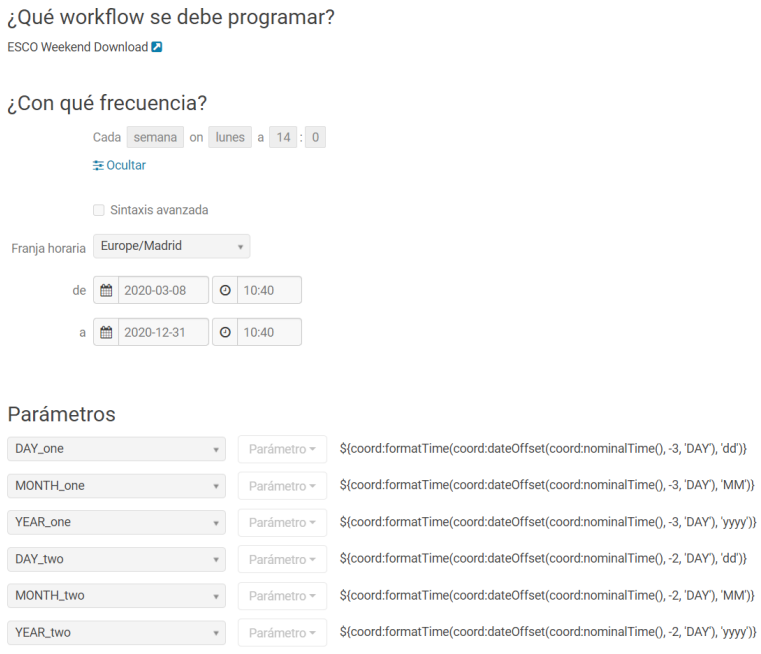


Figura 5.5: Coordinador para la extracción y procesado de los datos de ESCO del fin de semana

Cada uno de estos coordinadores ejecuta un workflow, que es un gráfico acíclico que representa una serie de trabajos a ejecutar de manera automatizada mediante *Oozie*. Estos workflows pueden contener cualquier trabajo que se pueda ejecutar en el entorno Hadoop, como consultas Hive, scripts de PySpark, Java, Shell, etc. A continuación se muestran y explican los esquemas de los tres workflows ejecutados en los tres coordinadores:

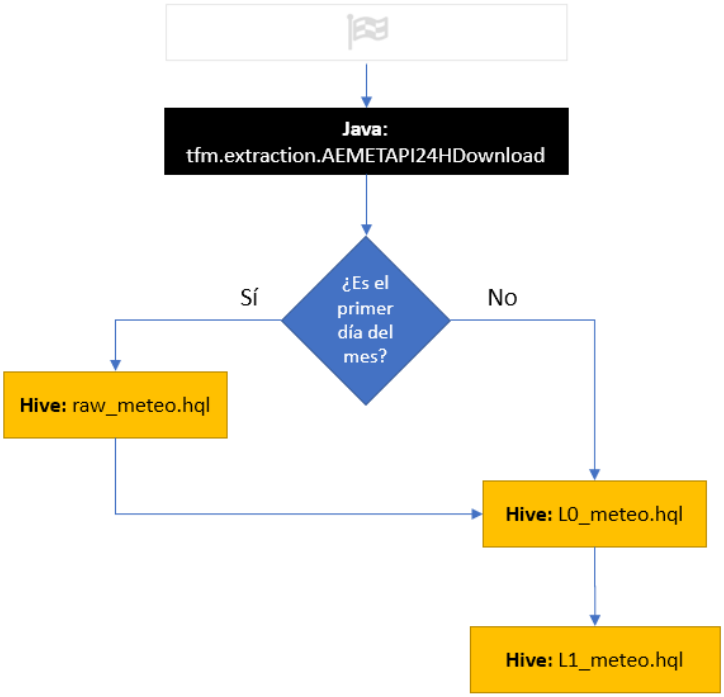


Figura 5.6: Workflow Daily AEMET Download

- **Java: tfm.extraction.AEMETAPI24HDownload**: Este trabajo ejecuta el script de Java que descarga y almacena en HDFS los datos desde la API de AEMET.
- Nodo de decisión y **raw_meteo**: Se comprueba si el parámetro introducido ${DAY} es igual a '01', en cuyo caso es necesario crear una nueva partición para los datos brutos de meteorología utilizando el script raw_meteo.hql:

Listado de código 5.7: Contenido del script raw_meteo.hql

```
ALTER TABLE meteo ADD PARTITION (yymm='${YEAR}${MONTH}') LOCATION '/user/
    esther/TFM/raw/AEMET/${YEAR}/${MONTH}';
```

- **L0_meteo**: Si no es el primer día del mes, se pasa directamente a este script, en caso contrario se llegaría tras ejecutar raw_meteo. Este script realiza la tarea de *Alignment* diseñada en 4.3, asegurándonos de que no se incluyen datos duplicados en la tabla. Para ello utiliza el siguiente código:

Listado de código 5.8: Contenido del script L0_meteo.hql

```
INSERT INTO l0_meteo
SELECT idema as idEstacion,
  to_date(from_unixtime(unix_timestamp(fint,"yyyy-MM-dd'T'HH:mm:ss"))) as
     fecha,
  hour(from_unixtime(unix_timestamp(fint,"yyyy-MM-dd'T'HH:mm:ss"))) as hora,
  MIN(ta) as temperatura, MIN(vv) as velocidad_viento, MIN(dv) as
     direccion_viento,
  MIN(vmax) as velocidad_racha, MIN(dmax) as direccion_racha,
  MIN(prec) as precipitacion, MIN(pres) as presion, MIN(hr) as humedad
FROM meteo
GROUP BY idema,
  to_date(from_unixtime(unix_timestamp(fint,"yyyy-MM-dd'T'HH:mm:ss"))),
  hour(from_unixtime(unix_timestamp(fint,"yyyy-MM-dd'T'HH:mm:ss")))
HAVING to_date(from_unixtime(unix_timestamp(fint,"yyyy-MM-dd'T'HH:mm:ss"))) =
   to_date('${YEAR}-${MONTH}-${DAY}');
```

- **L1_meteo**: Finalmente se ejecuta el script que realiza la tarea de *Enrichment* diseñada en 4.3. Se ejecuta la siguiente consulta:

Listado de código 5.9: Contenido del script L1_meteo.hql

```
INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L1/estacion_meteo'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
STORED AS TEXTFILE

SELECT DISTINCT idema, ubi, lon, lat, alt
FROM meteo;
```

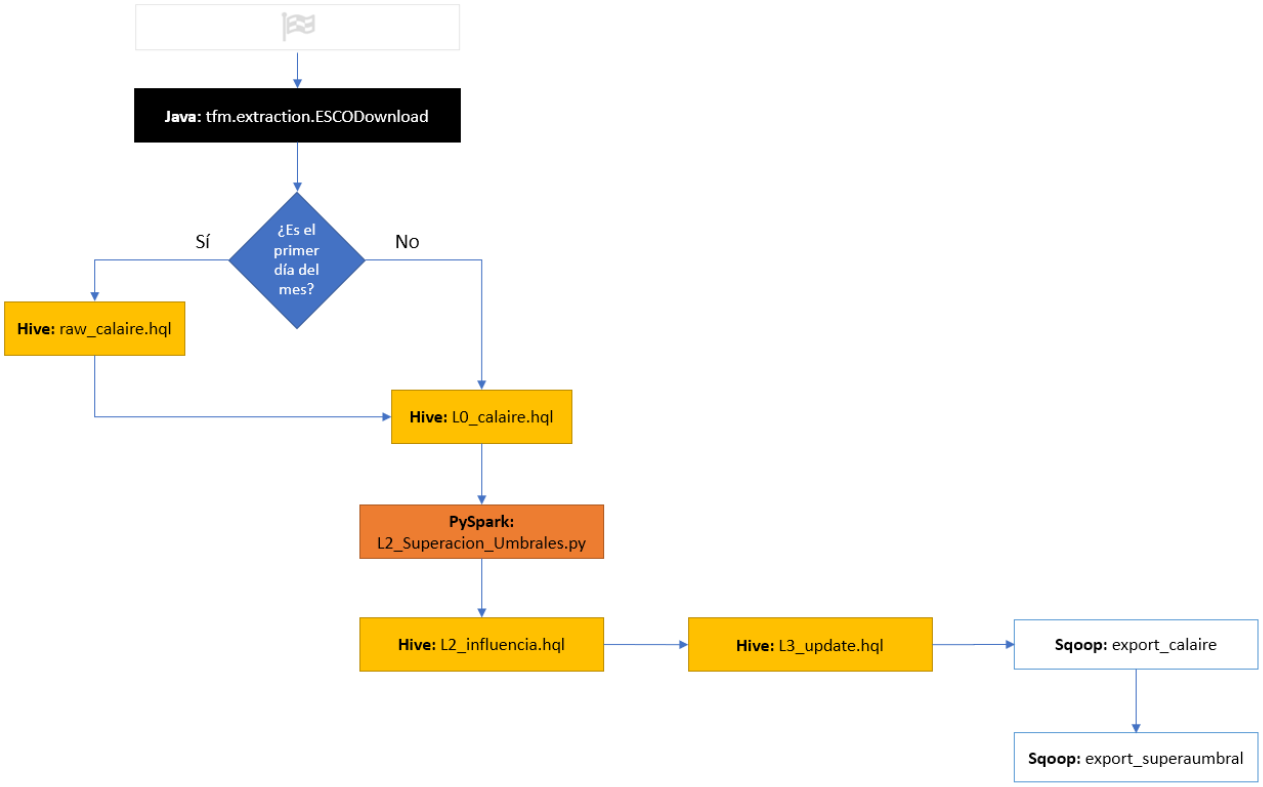


Figura 5.7: Workflow ESCO Download

- **Java: tfm.extraction.ESCODownload**: Ejecuta el programa que descarga y almacena en HDFS los datos del portal de Control de la Calidad del Aire de la Junta.
- Nodo de decisión y **raw_calaire**: De manera análoga al workflow de AEMET, se debe crear una nueva partición para los datos en bruto de contaminantes si es el primer día del mes:

Listado de código 5.10: Contenido del script raw_calaire.hql

```
ALTER TABLE calaire ADD PARTITION (yymm='${YEAR}${MONTH}') LOCATION '/user/
    esther/TFM/raw/ESCO/${YEAR}/${MONTH}';
```

- **L0_calaire**: También análogo al workflow anterior, realiza el *Alignment* de los datos de contaminación 4.3.

Listado de código 5.11: Contenido del script L0_calaire.hql

```
INSERT INTO l0_calaire
SELECT idEstacion,
  to_date(from_unixtime(unix_timestamp(fecha, 'dd/MM/yyyy␣HH:mm:ss'))) as
      fecha,
  hour(from_unixtime(unix_timestamp(fecha, 'dd/MM/yyyy␣HH:mm:ss'))) as hora,
```

```
    MIN(co) as co, MIN(no) as no, MIN(no2) as no2, MIN(o3) as o3, MIN(pm10) as
        pm10,
    MIN(pm25) as pm25, MIN(so2) as so2
FROM calaire GROUP BY idEstacion, to_date(from_unixtime(unix_timestamp(fecha,
    'dd/MM/yyyy␣HH:mm:ss'))),
    hour(from_unixtime(unix_timestamp(fecha, 'dd/MM/yyyy␣HH:mm:ss')))
HAVING to_date(from_unixtime(unix_timestamp(fecha, 'dd/MM/yyyy␣HH:mm:ss'))) =
    to_date('${YEAR}-${MONTH}-${DAY}');
```

- **L2_Superacion_Umbrales.py**: En esta etapa se realiza el procesamiento de los datos de contaminantes para detectar posibles superaciones de los umbrales de concentración de polución enmarcada en la etapa de *Integration* 4.3. Debido a la complejidad de este proceso se ha implementado mediante PySpark en lugar de Hive. Las superaciones detectadas se almacenan en la ruta /user/esther/TFM/L2/superaumbral desde dicho script.

- **L2_influencia.hql**: Con este script de Hive se realiza la segunda tarea de *Integration* en la que se crea la relación entre los datos de contaminantes y meteorología alineados anteriormente. Para ello se ejecuta:

Listado de código 5.12: Contenido del script L2_influencia.hql

```
INSERT INTO L2_influencia
SELECT m.fecha as fecha_meteo, m.hora as hora_meteo,
  m.idestacion as idEstacion_meteo, c.fecha as fecha_calaire,
  c.hora as hora_calaire, c.idestacion as idEstacion_calaire,
  !(f.fecha IS NULL) as festivo
FROM I0_calaire c LEFT OUTER JOIN I0_festivos f ON c.fecha=f.fecha
  LEFT OUTER JOIN I0_meteo m ON (c.fecha=m.fecha AND c.hora=m.hora)
WHERE to_date(c.fecha) = to_date('${YEAR}-${MONTH}-${DAY}') AND
  to_date(m.fecha) = to_date('${YEAR}-${MONTH}-${DAY}');
```

- **L3_update.hql**: Este el último script ejecutado en Hive y realiza la desnormalización de los datos, la actualización de los datos refinados y la recreación las tablas temporales que serán cargadas en el entorno de explotación en las próximas dos etapas.

Listado de código 5.13: Contenido del script L3_update.hql

```
INSERT OVERWRITE TABLE DH_EstacionMeteo SELECT * FROM I1_estacion_meteo;

INSERT INTO TABLE DH_MedicionMeteo
SELECT *
FROM I0_meteo
WHERE to_date(fecha) = to_date('${YEAR}-${MONTH}-${DAY}');

INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L3/
    temp_medicioncalaire'
```

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
    AS TEXTFILE
SELECT c.idestacion, c.fecha, c.hora, c.co, c.no, c.no2, c.o3, c.pm10, c.pm25,
    c.so2,
        m.temperatura, m.velocidad_viento, m.direccion_viento, m.
            velocidad_racha,
        m.direccion_racha, m.precipitacion, m.presion, m.humedad, f.festivo
FROM I0_calaire c LEFT JOIN I2_influencia f ON
    (f.fecha_calaire=c.fecha AND f.hora_calaire=c.hora AND f.
        idestacion_calaire=c.idestacion)
  LEFT OUTER JOIN I0_meteo m ON
    (f.fecha_meteo=m.fecha AND f.hora_meteo=m.hora AND f.idestacion_meteo=m.
        idestacion)
WHERE to_date(c.fecha) = to_date('${YEAR}-${MONTH}-${DAY}');

INSERT INTO TABLE DH_MedicionCalAire
SELECT *
FROM temp_medicioncalaire;

INSERT INTO TABLE DH_Influencia
SELECT *
FROM I2_influencia
WHERE to_date(fecha_calaire) = to_date('${YEAR}-${MONTH}-${DAY}') OR
  to_date(fecha_meteo) = to_date('${YEAR}-${MONTH}-${DAY}');

INSERT OVERWRITE DIRECTORY '/user/esther/TFM/transformation/L3/
    temp_superaumbral'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED
    AS TEXTFILE
SELECT su.*
FROM I2_superaumbral su LEFT JOIN I0_umbral u ON (u.id=su.idumbral)
WHERE (u.temporalidad <> 'anual' AND to_date(su.fecha)=to_date('${YEAR}-${
    MONTH}-${DAY}'))
  OR (u.temporalidad = 'anual' AND ${MONTH}='01' AND ${DAY}='01' AND ${YEAR}=
      CAST(year(su.fecha)-1 AS STRING));

INSERT INTO TABLE DH_SuperaUmbral SELECT * FROM temp_superaumbral;
```

- Las dos últimas etapas, **export_calaire** y **export_superaumbral** son trabajos de Sqoop, y llevan a cabo la carga de las tablas *temp_medicioncalaire* y *temp_superaumbral* en el entorno de explotación. Se describirán las tareas de estas dos etapas en el apartado5.3.

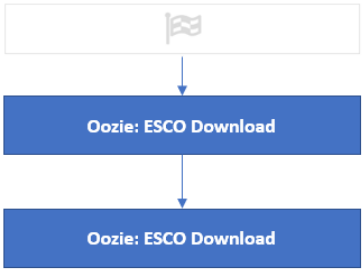


Figura 5.8: Workflow ESCO Weekend Download

Para el workflow que ejecuta la descarga y procesamiento de los datos del Viernes y Sábado se aprovecha la capacidad de Oozie de ejecutar workflows dentro de otros, simplemente ejecutando el mismo proceso que se ha mencionado anteriormente dos veces, una para el Viernes y otra para el Sábado.

5.3. Carga

La carga es el proceso por el cual los datos transformados pasan al entorno de explotación donde son utilizados para visualización, análisis y predicción. Como ya se ha descrito en la sección 5.3, este proceso se lleva a cabo con la herramienta Apache Sqoop. Este programa es bastante sencillo, y permite con un solo comando realizar transferencias bidireccionales entre Hadoop y bases de datos relacionales. El comando utilizado en nuestro caso es el siguiente:

Listado de código 5.14: Comando ejemplo para transferir datos de nuestro cluster Hadoop a la base de datos en el entorno de explotación

```
sqoop export --connect jdbc:sqlserver://10.124.5.22:1433 --username cloudera --
    password clouderalogin --export-dir /user/esther/TFM/transformation/L3/
    temp_medicioncalaire --table MedicionCalAire --input-null-string "\\\\N" --
    input-null-non-string "\\\\N"
```

El comando tiene las siguientes partes:

1. sqoop export: indica al programa sqoop que la operación a realizar es una de **exportación** desde Hadoop a el sistema destino.
2. --connect jdbc:sqlserver://10.124.5.22:1433: permite conectarse a la base de datos mediante un conector jdbc:sqlserver, compatible con Microsoft SQL Server, en la dirección IP 10.124.5.22, que es la dirección pública del entorno de explotación, y el puerto 1433, por defecto en SQL Server.
3. --username cloudera --password clouderalogin: indica las credenciales para conectarse a un usuario del servidor con permisos para escritura en la base de datos TFM.

4. ––export–dir /user/esther/TFM/transformation/L3/temp_medicioncalaire ––table MedicionCalAire: señala el directorio HDFS que contiene los datos a exportar así como la tabla de la base de datos donde se añadirán los datos.

5. ––input–null–string "\\\\N" ––input–null–non–string "\\\\N": por último, es necesario indicarle a sqoop que el símbolo «\N» corresponde a un valor vacío o NULL, y que debe trasladarse a la tabla de SQL Server como tal.

Ya que el entorno de explotación está preparado para usar mediciones de una estación meteorológica, podemos prescindir de algunas tablas del Refined Data, específicamente de *MedicionMeteo*, ya que se han desnormalizado los datos en la tabla *MedicionCalAire*; *EstacionMeteo,* ya que solo existe una estación de meteorología; e *Influencia*, debido a que gracias a la desnormalización no es necesario almacenar la relación entre mediciones meteorológicas y de calidad del aire.

Previo a la carga de los datos, de similar manera a la transformación, es necesario crear las tablas en la base de datos del SQL Server. Para ello se ha ejecutado el código presente en el Apéndice B.

Capítulo 6

Exploración

El planteamiento inicial del proyecto es el de utilizar los datos almacenados en el DataHub para crear una visualización de datos destinada a expertos en calidad del aire, tratando de responder a la necesidad de **controlar los indicadores de calidad del aire capturados cada día en las estaciones de calidad del aire de Valladolid,** así como **explorar la evolución de la calidad del aire en Valladolid durante cierto rango de fechas**. En este capítulo se lleva a cabo una planificación completa de dicha visualización así como la descripción de su implementación final.

6.1. Planificación

Se ha seguido la metodología de diseño de visualizaciones propuesta por *Data Visualization: A Successful Design Process* por Andy Kirk [34], que establece una serie de etapas y consideraciones a tener en cuenta en un proyecto de visualización.

6.1.1. Propósito

El primer paso en la metodología es establecer los tres aspectos necesarios para definir el propósito del proyecto:

- La **motivación**, definida por los usuarios objetivo y sus necesidades.
- La **intención**, marcada por la función y tono de la visualización.
- El impacto de otros **factores clave** que afectan al proyecto, principalmente las restricciones y requerimientos.

Motivación La visualización tiene como objetivo que un experto en calidad del aire tenga una visión completa de los datos de contaminantes y meteorología recogidos durante el día anterior, de manera horaria y por estaciones,

y que además pueda realizar una comparación entre los datos actuales y el histórico de datos recogidos, con un rango temporal de longitud variable.

Algunas de las preguntas a las que podrá responder esta visualización son:

- ¿Qué estaciones presentaron ayer unos valores más altos de sus contaminantes medidos, en comparación con los datos del resto del año actual?
- ¿Cómo fue la velocidad de las rachas de viento en Valladolid durante la noche anterior? ¿Coinciden las horas de mayor velocidad con una bajada en los datos de PM10 registrados?
- ¿Cual fue el máximo valor de NO2 del día anterior? ¿Superó alguno de los umbrales?
- ¿Qué estación tuvo el mayor valor promedio de NO2? ¿Es este valor alto comparado con valores históricos de NO2 recogidos en esta estación?
- ¿Sigue la evolución horaria de CO recogido en la estación Renault 2 una tendencia normal comparada con los datos que se suelen registrar en esa estación? (por ejemplo niveles más altos durante la hora de entrada de trabajadores, y más bajos durante la noche)
- ¿Hay una mejora significativa de la calidad del aire durante 2020 en la zona controlada por la estación Arco de Ladrillo I, en comparación con los datos registrados por dicha estación durante 2018?
- ¿Causaron las medidas de confinamiento por el COVID-19 en Valladolid un descenso significativo en los contaminantes respecto a las medidas registradas los meses de Marzo, Abril y Mayo de años anteriores?

Intención La función de la visualización es exploratoria, ya que no existe una narrativa que el diseñador busque transmitir con la visualización, si no que la intención es proporcionar herramientas para que el usuario pueda familiarizarse con los datos y explorarlos.

El tono de la visualización será pragmático, debido a que su público no necesita ser «motivado» sobre la calidad del aire. Se centrará por lo tanto en la precisión y utilidad analítica de la representación.

Factores clave Las especificaciones de la visualización surgen de la exploración de principalmente dos herramientas de visualización de este tipo de datos: la propia que ofrece en su portal el Ayuntamiento de Valladolid[1], y la disponible en el portal del World Air Quality Index Project[2].

Respecto al portal del Ayuntamiento de Valladolid, este ofrece los indicadores por contaminante o por estación y con dos granularidades temporales distintas: las 24 horas anteriores, o datos instantáneos. En el caso de las 24h anteriores, muestra tanto la evolución como los principales estadísticos: valores medio, mínimo y máximo del día.

[1] https://www.valladolid.es/en/rccava/datos-red/
[2] https://aqicn.org/

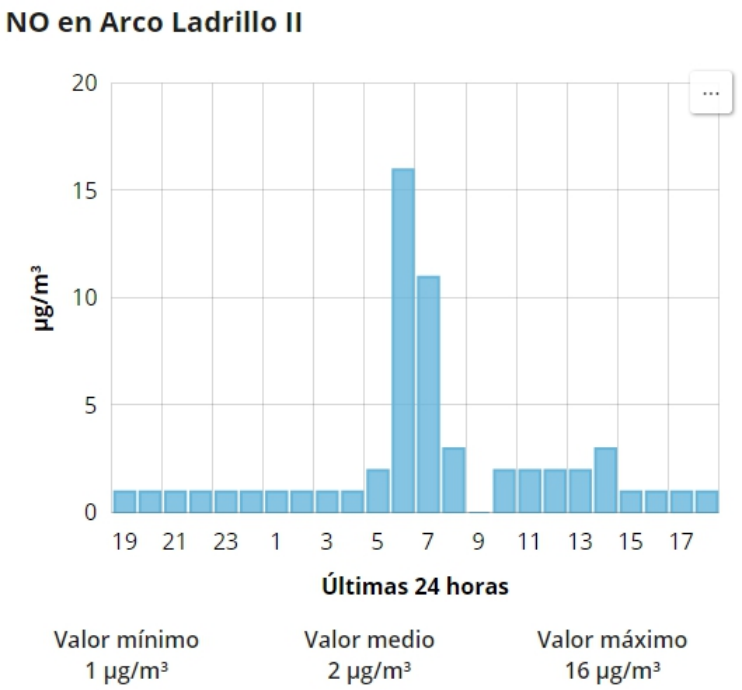


Figura 6.1: Visualización de los datos de las últimas 24h de NO en Arco de Ladrillo II por el portal del Ayuntamiento de Valladolid - Fuente: https://www.valladolid.es/es/rccava/datos-red/datos-ultimas-24-horas

Respecto a los datos instantáneos, se ofrece el dato contextualizado por un mínimo, un máximo y un valor medio, así como la última actualización del dato, y una comparativa entre este dato y el de días anteriores (se desconoce si el dato de días anteriores se refiere a la misma hora del dato actual, media diaria u otra agregación).

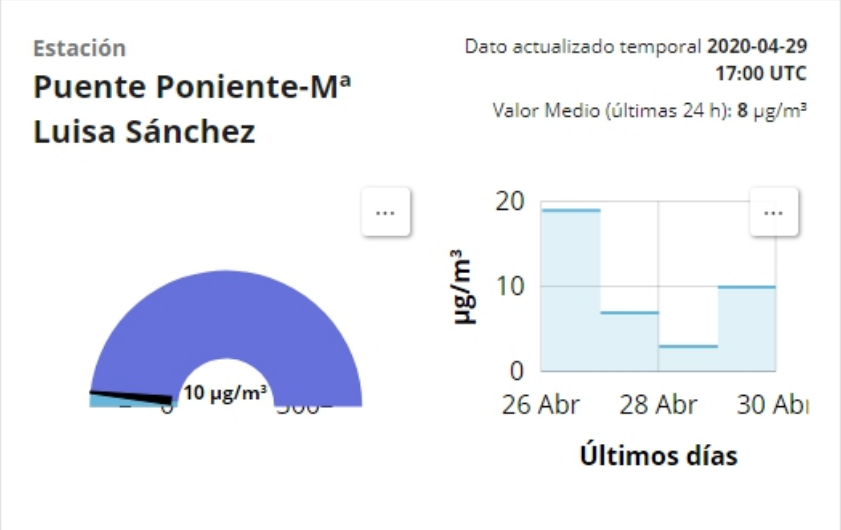


Figura 6.2: Visualización de los datos instantáneos de PM10 en Puente Poniente por el portal del Ayuntamiento de Valladolid - Fuente: https://www.valladolid.es/es/rccava/datos-red/datos-actualizados-temporales

De estos gráficos se ha tratado de imitar su sencillez, e incorporar los siguientes requerimientos:

- Los datos estarán separados por estación y contaminante.
- Se ofrecerán los datos horarios para las últimas 24h, así como los valores medio, máximo y mínimo del día.
- Los datos se contextualizarán, ofreciendo un rango de valores que tendrá que ver con el comportamiento en el pasado de dicho contaminante en dicha estación.

Por otro lado, estos gráficos carecen de una leyenda más documentada, lo que deja muchas dudas sobre el significado de ciertas medidas representadas. Son completamente estáticos, y el mayor histórico disponible son 24 horas, lo que limita mucho la flexibilidad de los mismos para un análisis extenso.

Respecto al portal del World Air Quality Index Project, se ofrece un panel para cada una de las estaciones registradas, en este caso de todo el mundo, incluyendo datos de contaminantes, las principales variables meteorológicas y un índice de calidad que permite conocer el estado general de la zona de un solo vistazo:

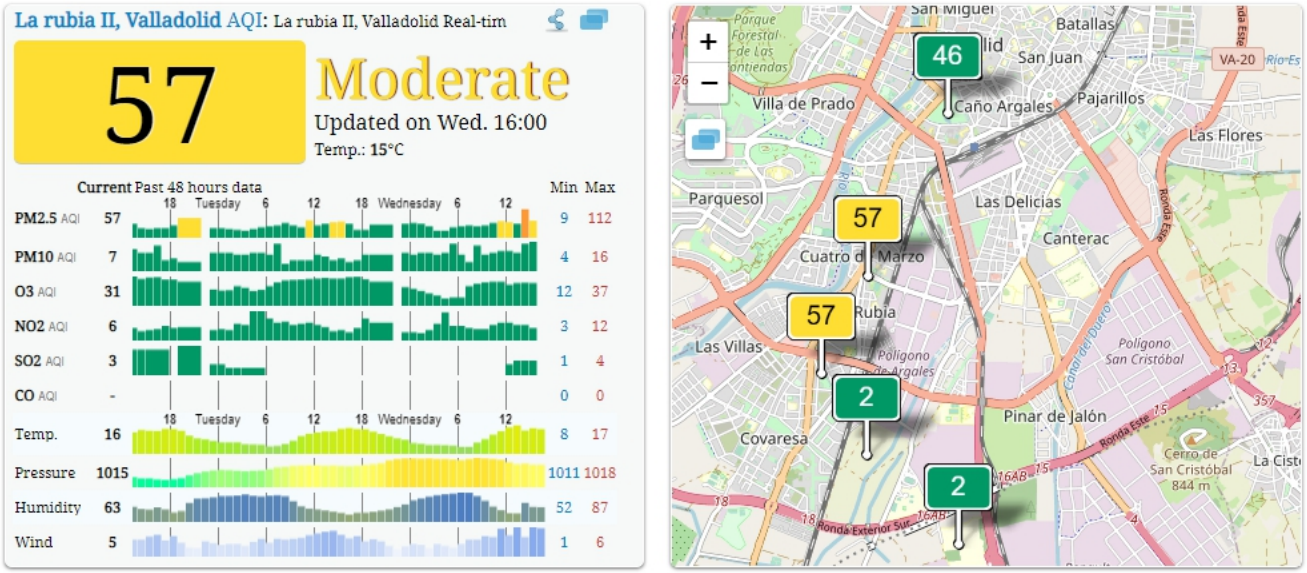


Figura 6.3: Visualización de datos en La Rubia II por el portal de World Air Quality Index Project - Fuente: https://aqicn.org/city/spain/castilla-y-leon/valladolid/arco-de-ladrillo-ii/

De este gráfico se ha tratado de imitar su completitud y diseño compacto, siendo capaz de mostrar una gran cantidad de información en un espacio pequeño. Inspirado por esta visualización, se incorporarán los siguientes requerimientos:

- Debe ser posible observar, de un vistazo, el estado actual del aire medido por cada estación.
- Se ofrecerán datos de las principales variables meteorológicas, de tal modo que su evolución temporal se pueda comparar con la de los contaminantes del aire.

De nuevo se cuenta con un gráfico estático y con una temporalidad limitada a 48h, limitando la usabilidad para análisis complejos. Los valores se han convertido de sus unidades originales a un valor desarrollado por la herramienta, que llaman «AQI», lo cual es útil para comparar entre países que utilizan distintos estándares, pero puede confundir a un experto que conozca el significado de las unidades reales.

6.1.2. Enfoque

Uno de los enfoques clave tomados para nuestra visualización es la capacidad del usuario de «modificarla» para responder al mayor rango de preguntas posible, a la vez que se siguen los principios necesarios para que el resultado sea fácil de usar, intuitivo, y atractivo. Esto nos lleva a la importancia de utilizar una herramienta de creación de visualizaciones dinámicas.

Además se buscará potenciar las siguientes dimensiones:

- La **tendencia** temporal de los datos, en este caso con granularidad horaria.
- Las **comparaciones** entre los datos recogidos en distintas estaciones, así como entre momentos temporales relacionados.
- La **relación** entre las distintas variables, con especial enfoque en la conexión entre datos de contaminantes y de meteorología.

6.1.3. Diseño

En esta etapa se concretan detalles como los datos a usar y la presentación más adecuada a lo planteado en fases anteriores.

Datos a usar Partiendo de los datos refinados y almacenados en el DataHub, para estas visualizaciones serán necesarias las tablas:

- **MedicionCalAire**: La fecha, hora y estación de recogida de datos de cada contaminante, así como las variables meteorológicas para dicha fecha y hora en Valladolid.
- **EstacionCalAire**: El catálogo de las estaciones de medida de contaminantes en Valladolid, específicamente su nombre y sus coordenadas.
- **Umbral**: El inventario de umbrales definidos con el contaminante, la temporalidad, y el valor límite.
- **SuperaUmbral**: Cada una de las mediciones de calidad del aire que superaron alguno de los umbrales, con información sobre el valor con el que se superó dicho límite.

Presentación Se ha decidido desarrollar una visualización tipo dashboard para consolidar la metáfora de un cuadro de mandos para el control de la calidad del aire en Valladolid. En un dashboard se prioriza maximizar la información mostrada minimizando una excesiva densidad de datos que puede causar confusión o dificultad en su interpretación. Este tipo de visualizaciones deben poder ser exploradas de un vistazo.

6.2. Implementación

A continuación se detalla la implementación de la visualización siguiendo lo descrito en pasos anteriores.

6.2.1. Transformación de datos

Desde los datos obtenidos del DataHub, se realizan algunas modificaciones tras su importado a PowerBI, utilizando PowerQuery, el editor de datos de PowerBI. Estas transformaciones, se aplican automáticamente para los datos nuevos cuando estos se actualizan.

Se transforman las fechas en SuperaUmbral y MedicionCalAire de tipo Texto a tipo Fecha. Esta transformación es tan simple como designar dichas columnas como tipo Fecha.

Mientras que la mayoría de relaciones entre tablas se detectan automáticamente, la relación entre SuperaUmbral y MedicionCalAire no lo hace. Esto se debe a que la clave primaria de MedicionCalAire y foránea en SuperaUmbral se trata de una clave compuesta de varias columnas, lo cual no está soportado en PowerBI. Para remediarlo creamos una nueva columna en ambas tablas, mediante la fórmula DAX:

Listado de código 6.1: Fórmula para crear nuevos índices para relacionar las tablas SuperaUmbral y MedicionCalAire

```
Text.Combine({Text.From([idEstacion], "es-ES"), ";",
                        Text.From([fecha], "es-ES"), ";",
                        Text.From([hora], "es-ES")}
```

Esta columna actuará como identificador de la relación, y que se trata de la concatenación de las claves tal que el resultado es: *idEstacion;fecha;hora*. Esta columna sigue siendo única en MedicionCalAire, y se puede utilizar como clave en la relación de las dos tablas.

Se deseaba añadir marcadores de tipo punto en los gráficos de línea y área (por ejemplo en la evolución horaria de los contaminantes) para poder ocultar el eje X sin eliminar completamente la información de en qué hora se produce cada dato. PowerBI no permite que se muestren dichos marcadores cuando el eje X es de tipo continuo, lo que es automáticamente determinado cuando la variable en el eje X es numérica, como es el caso de la variable hora de tipo Entero. Por tanto, se quería cambiar el eje X a un eje de tipo categórico utilizando una variable de tipo Texto. A parte de convertir la columna «hora» a tipo Texto, se ha añadido un 0 delante de las horas del 0 al 9, para que la ordenación de las mismas se haga correctamente, utilizando la fórmula DAX:

Listado de código 6.2: Fórmula para extender la hora a dos dígitos

```
Text.PadStart(Text.From([hora], "es-ES"), 2, "0")
```

Para la comparación entre datos actuales y datos históricos deben utilizarse medidas para cada contaminante que no se vean afectadas por filtros sobre la fecha de datos actuales, pero sí sobre la estación de recogida y la hora, para poder representarlos de manera horaria y que cambien si se selecciona una u otra estación. Inicialmente se crean estas medidas con el valor medio para todos los datos de MedicionCalAire disponibles.

Listado de código 6.3: Fórmula para calcular una media horaria histórica de los datos de Ozono por estación de recogida

```
O3_MediaHistorica =
        CALCULATE(
                AVERAGEX(MedicionCalAire;MedicionCalAire[O3]);
                ALLEXCEPT(MedicionCalAire;EstacionCalAire[nombre];
                        MedicionCalAire[hora_str])
        )
```

Para la tabla resumen por estaciones se crean dos medidas por cada contaminante, conteniendo el valor de los percentiles 33 % y 66 % calculados sobre todos los datos (de manera similar a la anterior, excepto sólo manteniendo el filtro de estación). Por ejemplo:

Listado de código 6.4: Fórmula para calcular el percentil 33 % del histórico de Óxido de Nitrógeno por estación de recogida

```
NO_1stPercentile =
        CALCULATE(
                PERCENTILE.INC(MedicionCalAire[NO];0,33);
                ALLEXCEPT(MedicionCalAire;EstacionCalAire[nombre])
)
```

. También se crea una nueva columna por cada contaminante en la tabla de MedicionCalAire, que contiene, para cada fila, en qué tercil se sitúa la medida de contaminante (0, 1 o 2) según si su valor es menor que el percentil 33 %, está entre los dos percentiles, o es superior al percentil 66 % para ese contaminante.

Listado de código 6.5: Fórmula para calcular en qué tercil se situa cada valor de Óxido de Nitrógeno

```
IF(ISBLANK([NO]);BLANK();
        IF([NO]<=[NO_1stPercentile];0;
                IF([NO]<=[NO_2ndPercentile];1;2))
```

Para permitir que el usuario controle las fechas sobre las que se calculan los datos de referencia mencionados anteriormente, se siguen los siguientes pasos:

1. Se crea una nueva tabla: FechaHistorico. Esta tabla contendrá una sola columna con una lista de fechas que irán desde el primer día de 2018 (primer dato disponible de MedicionCalAire) a la fecha anterior al día actual. Se crean también dos medidas en la misma tabla, que contienen el mínimo y el máximo de la columna. El filtro de fechas históricas se aplicará sobre la columna de esta tabla, con la actualización automática de ComienzoRango y FinalRango.

Listado de código 6.6: Fórmula para crear una fila con cada día entre el 1 de Enero de 2018 al día actual

```
FechaHistorico = GENERATESERIES(DATE(2018;01;01);TODAY()-1;1)
```

Listado de código 6.7: Fórmula para calcular la primera y última fecha en la columna anterior

```
ComienzoRango = MIN(FechaHistorico[Fechas])
FinalRango = MAX(FechaHistorico[Fechas])
```

2. Se duplica la columna fecha de la tabla MedicionCalAire a fecha_aux. Esto es necesario ya que no podemos ignorar un filtro y filtrar la misma columna en la misma función DAX.

3. Se actualizan las medidas históricas para cada contaminante así como 1stPercentile y 2ndPercentile para añadir un filtro adicional de que la fecha de datos debe estar entre las medidas ComienzoRango y FinalRango

Listado de código 6.8: Fórmula para calcular una media horaria entre dos fechas de los datos de Ozono por estación de recogida

```
O3_MediaHistorica =
CALCULATE(
 AVERAGEX(MedicionCalAire;MedicionCalAire[O3]);
  FILTER( ALLEXCEPT(MedicionCalAire;EstacionCalAire[nombre];
   MedicionCalAire[hora_str]);
   MedicionCalAire[fecha_aux].[Date]>=FechaHistorico[ComienzoRango] &&
   MedicionCalAire[fecha_aux].[Date]<=FechaHistorico[FinalRango] )
)
```

Se busca adaptar un gráfico tipo radar a mostrar la velocidad mayoritaria por cada dirección cardinal. Para ello nos encontramos con dos problemas: (i) la variable dirección no siempre contiene un ángulo exacto correspondiente a las 8 direcciones cardinales (Norte, Noreste, Este, Sudeste, Sur, Sudoeste, Oeste y Noroeste), (ii) para que la posición de cada dirección tenga sentido, se deben mostrar en todo momento todas las direcciones posibles (desde 0^{o} hasta 315^{o}, una cada 45^{o}), sin importar si dichas direcciones han ocurrido alguna vez. Para poder elaborar correctamente este gráfico se realizan las siguientes transformaciones:

1. Se crean dos nuevas columnas en la tabla MedicionCalAire que contendrán una simplificación de los ángulos de dirección de viento y racha, convirtiéndolos en uno de los 8 valores posibles según cual esté más cercano al valor real.

Listado de código 6.9: Fórmula para simplificar la dirección de racha al ángulo simplificado más cercano

```
direccion_racha_simplificada =
  SWITCH(TRUE(),
    ISBLANK(MedicionCalAire[direccion_racha]), BLANK(),
      MedicionCalAire[direccion_racha]<45 &&
        ABS(MedicionCalAire[direccion_racha]-0) <
                ABS(MedicionCalAire[direccion_racha]-45), 0,
      MedicionCalAire[direccion_racha]<90 &&
        ABS(MedicionCalAire[direccion_racha]-45) <
                ABS(MedicionCalAire[direccion_racha]-90), 45,
      MedicionCalAire[direccion_racha]<135 &&
        ABS(MedicionCalAire[direccion_racha]-90) <
                ABS(MedicionCalAire[direccion_racha]-135), 90,
      MedicionCalAire[direccion_racha]<180 &&
```

```
  ABS(MedicionCalAire[direccion_racha]-135) <
          ABS(MedicionCalAire[direccion_racha]-180), 135,
MedicionCalAire[direccion_racha]<225 &&
  ABS(MedicionCalAire[direccion_racha]-180) <
          ABS(MedicionCalAire[direccion_racha]-225), 180,
MedicionCalAire[direccion_racha]<270 &&
  ABS(MedicionCalAire[direccion_racha]-225) <
          ABS(MedicionCalAire[direccion_racha]-270), 225,
MedicionCalAire[direccion_racha]<315 &&
  ABS(MedicionCalAire[direccion_racha]-270) <
          ABS(MedicionCalAire[direccion_racha]-315), 270,
MedicionCalAire[direccion_racha]<360 &&
  ABS(MedicionCalAire[direccion_racha]-315) <
          ABS(MedicionCalAire[direccion_racha]-360), 315,
MedicionCalAire[direccion_racha]=360 ||
  ABS(MedicionCalAire[direccion_racha]-315) >
          ABS(MedicionCalAire[direccion_racha]-360), 0 )
```

2. Se crea dos nuevas tablas: PosiblesDireccionesViento y PosiblesDireccionesRacha que contendrán 8 filas, una por cada posible dirección (del 0 al 315 con un valor cada 45 grados). Esta columna será la clave primaria de la relación de cada una de estas tablas con MedicionCalAire, utilizando la columna de dirección simplificada como clave foránea de la relación.

3. En los objetos visuales de tipo radar se puede utilizar como categoría la columna de la tabla de Posibles-Direcciones, y como Eje Y la suma del valor de velocidad de viento o racha respectivamente. Con la opción de «mostrar elementos sin datos» en el eje de categoría aparecen todas las direcciones posibles, así como un valor (Blank) en la parte superior del gráfico. Para remediarlo, se añade un filtro a nivel de objeto visual que elimina las filas en las que dirección está en blanco.

A la hora de realizar gráficos que muestren el estado del contaminante frente a los umbrales de calidad del aire establecidos, se ha optado por utilizar el objeto visual Medidor, al ser consonante con el estilo del resto de la visualización, gratuito, y mostrar de una manera clara el "progreso" del contaminante hacia los umbrales. Para utilizar este gráfico correctamente se ha creado una medida por cada umbral que nos interesa. En este caso, ya que el dashboard tiene carácter diario, se crea una medida por cada uno de los 7 umbrales diarios y horarios existentes. La creación de esta medida es manual, sin embargo no se anticipa actualización de dicha tabla. Estas medidas son las que se utilizan como valores límite o máximo en el Medidor.

Listado de código 6.10: Fórmula para crear una medida con el umbral de Dióxido de Nitrógeno horario

```
UmbralNO2_horario = CALCULATE( MIN(Umbral[limite]), Umbral[contaminante] = "NO2",
    Umbral[temporalidad] = "horaria"
```

6.2.2. Detalle de la visualización

En esta sección se muestran capturas de pantalla de la visualización final, así como la justificación detrás de cada elemento seleccionado.

La visualización se divide en tres partes: filtros que permiten al usuario controlar los datos mostrados; datos de calidad del aire que muestran los datos recogidos por hora para cada uno de los contaminantes medidos en Valladolid, así como un resumen del estado de cada estación; y datos de meteorología que muestran las variables meteorológicas disponibles en Valladolid, también con carácter horario.

Figura 6.4: Visualización para el control de calidad del aire en Valladolid

Filtros Los principales filtros que controlan la visualización son los filtros de fecha actual e histórica, y el filtro de estaciones. Mientras que PowerBI permite que el usuario de la visualización aplique filtros en cualquiera de los campos de datos disponibles, se ha decidido incorporar a la visualización solo estos tres, dándoles visibilidad como los filtros cruciales para el usuario final, permitiendo controlar la temporalidad de los datos, el periodo de referencia, y la estación de recogida.

El filtro de fechas actuales controla los valores principales mostrados en todas las gráficas. El filtro es de tipo

relativo, que permite filtrar a valores como «El último día de datos a partir de la fecha actual». El filtro de fechas históricas afecta a la línea comparativa (en gris) de las gráficas de calidad del aire, así como los valores respecto a los cuales se calcula la escala de color del resumen por estaciones. De igual manera se trata de un filtro relativo.

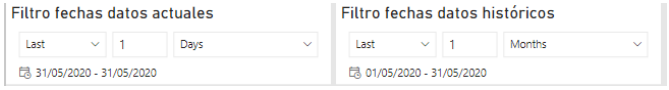


Figura 6.5: Filtros de fecha

El filtro de estaciones permite seleccionar una o más estaciones de calidad del aire, lo cual modificará los datos de calidad del aire, restringiendo tanto datos actuales como históricos a aquellos recogidos por dicha estación.

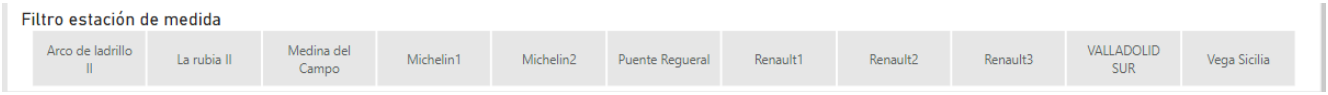


Figura 6.6: Filtro de estación

Además se han creado 5 marcadores en el dashboard, que permiten al usuario moverse rápidamente a algunas visualizaciones de ejemplo y dar más flexibilidad a los filtros de fechas:

- *Marcadores «Ayer contra el año anterior», «Ayer contra el mes anterior» y «Ayer contra la semana anterior».* Estos marcadores modifican el filtro de fechas histórico para seleccionar, en cada caso, los datos del año, mes o semana anteriores a la fecha actual.
- *Marcador «Ayer contra rango histórico libre»:* Este marcador cambia el filtro de fechas histórico de un filtro relativo a un rango de fechas, para así poder seleccionar libremente el inicio y final de fechas históricas comparativas.
- *Marcador «Rangos libres»:* Este marcador cambia los filtros de datos actuales y de fechas histórico de filtros relativos rangos de fechas, ofreciendo al usuario la máxima flexibilidad en la selección de datos.

Vale la pena mencionar que el usuario puede crear sus propios marcadores tras modificar los filtros, que aparecerán como marcadores personales en PowerBI.

Datos de calidad del aire Siguiendo los requerimientos mencionados en secciones anteriores, uno de los objetivos de la visualización era disponer de un resumen rápido del estado de la calidad del aire por estación. Para ello se ha optado por una visualización de tipo *heatmap*. Este tipo de visualización permite comparar de manera cruzada dos conjuntos de variables, en este caso estaciones y contaminantes. Muestra tres valores principales:

- El valor dentro de la casilla indica el valor promedio para dicho contaminante, recogido en la estación durante el periodo marcado por el filtro de fecha actual.
- El color de la casilla indica el tercil promedio en el que se sitúa el valor de dicho contaminante, en la estación. El tercil se calcula a partir de la distribución de datos marcados por el filtro de histórico, y puede ser 0 (bajo), 1 (medio) o 2 (alto). Cuanto más cercano a rojo, más alto (cercano a 2) es el tercil medio, y cuanto más cercano a verde, más bajo (cercano a 0).
- El icono indica el mayor tercil alcanzado durante el periodo actual, con verde siendo 0, amarillo siendo 1 y rojo siendo 2.

Por ejemplo se puede observar que la estación Michelin1 ha tenido valores promedio bajos para NO y NO2, muy altos para O3, y que además los valores máximos alcanzados son bajos en NO, medios en NO2 y altos en O3.

Esta tabla también indica el número de umbrales de calidad del aire superados por la estación durante el periodo de estudio.

Resumen por estaciones

Estación	NO	NO2	O3	PM10	PM2,5	CO	SO2	Umbrales superados
Arco de ladrillo II	1.6	8.6		12.9	9.8	0.3		
La rubia II	2.1	5.5		13.5	9.8		6.4	
Medina del Campo	1.0	5.1	72.7	22.0			3.0	
Michelin1	1.0	7.4	78.0					
Michelin2	1.0	6.0	66.3					
Puente Regueral	1.0	3.4	78.5	15.2	4.8			
Renault1								
Renault2								
Renault3								
VALLADOLID SUR			73.9					
Vega Sicilia	1.5	2.3	81.7	15.0				

Figura 6.7: Resumen por estaciones

Para mostrar la evolución diaria de cada contaminante, así como la comparación contra el histórico, se ha elegido un gráfico de líneas. Este gráfico suele usarse para mostrar valores cuantitativos sobre un periodo de tiempo, pudiendo analizar así la tendencia temporal de los datos. Además múltiples líneas permiten comparar de manera rápida dos series de datos, como es el caso entre los datos actuales y el histórico. Se ha elegido un color distinto para cada contaminante, y un gris para los datos históricos. El eje X muestra cada una de las horas, desde las 0h a las 23h, y el eje Y muestra el valor medio alcanzado en cada hora.

Además se muestran tres KPIs, un valor medio actual del contaminante, el valor mínimo y el valor máximo. Los colores elegidos (el color del contaminante para el promedio, y verde y rojo para mínimo y máximo respectivamente) refuerzan el significado de los valores indicado por las etiquetas.

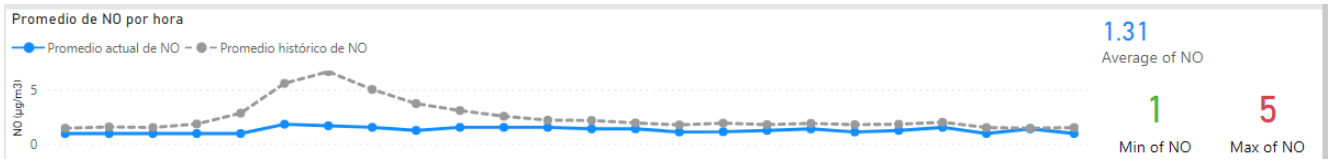


Figura 6.8: Evolución por hora de NO

En el caso de contaminantes con umbrales definidos, se ha optado por incorporarlos sustituyendo a los KPIs, utilizando un objeto tipo Medidor que muestra el progreso del valor actual para dichos KPIs (el máximo en el caso de umbrales horarios y el promedio en el caso de umbrales diarios) respecto a los umbrales definidos para dicho contaminante. Así, en el siguiente ejemplo se puede observar que el Ozono tiene dos umbrales horarios definidos, 180 y 240 $\mu g/m^3$, y que el valor actual se sitúa en 114 $\mu g/m^3$, por lo tanto a cierta distancia de ambos umbrales.

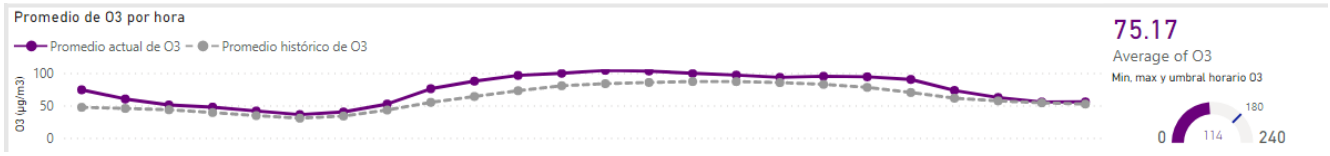


Figura 6.9: Evolución por hora de O3

Datos de meteorología En el caso de los datos de meteorología, nos interesa también tenerlos de manera horaria, sin embargo la importancia está en segundo plano respecto a los datos de calidad del aire, por lo que se le dedica un espacio más pequeño de la visualización.

Para las medidas con un solo valor por hora, como son la temperatura o la presión atmosférica se ha optado por un gráfico de área. Con una función similar al gráfico de línea, se ha elegido este tipo de visualización debido a un deseo de separar claramente su estilo de los gráficos de contaminantes.

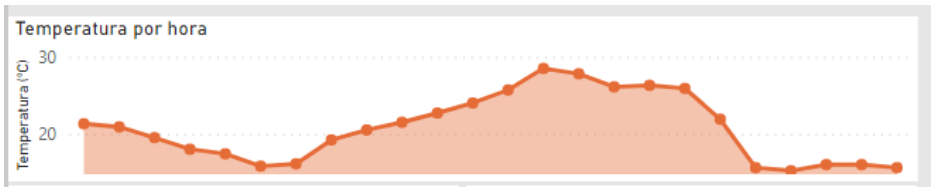


Figura 6.10: Evolución por hora de la temperatura

En el caso de la velocidad y dirección de viento y racha, se ha tratado de simplificar e implementar uno de las maneras más estandarizadas de mostrar datos de viento en el mundo de la meteorología, la rosa de vientos. Este gráfico muestra la dirección y velocidad del viento en un solo gráfico.

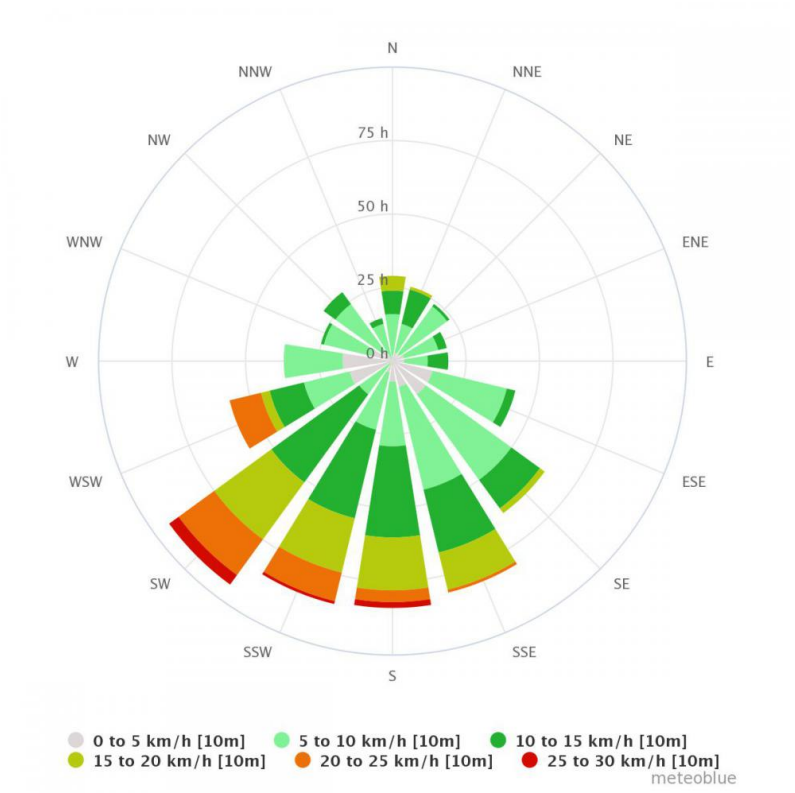


Figura 6.11: Ejemplo de un gráfico rosa de los vientos. Fuente: BREEZE Software / CC BY-SA

Debido a la complejidad de este gráfico y la ausencia de implementación en PowerBI, se ha optado por adaptar el gráfico de tipo radar[3]. Aunque algunas características del gráfico original han tenido que ser sacrificadas, el gráfico de radar permite observar la dirección predominante del viento utilizando la suma de las velocidades que se han observado en cada una de las direcciones. Así, por ejemplo, en el siguiente gráfico se puede observar que el viento ha soplado con más fuerza hacia el Oeste y Noroeste, con un valor medio hacia el Sur y Sudoeste y sin casi presencia en el resto de direcciones. Con intención de mostrar la velocidad desglosada de manera horaria se ha añadido un gráfico de líneas que contiene los valores de velocidad para viento y racha.

[3]Disponible en la tienda de componentes de PowerBI en la URL https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380771

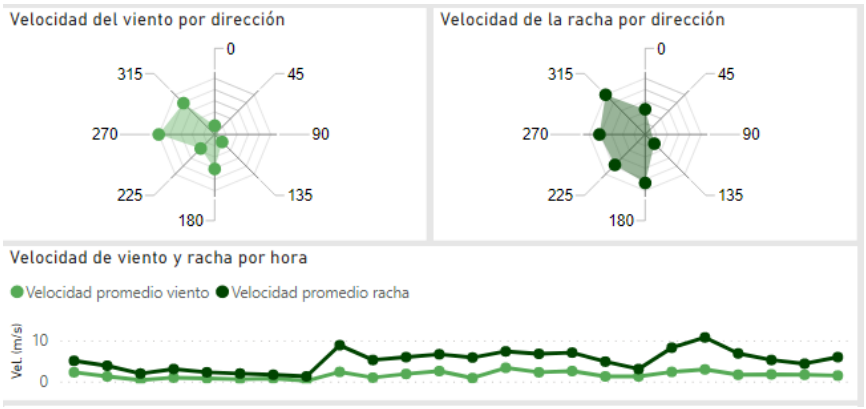


Figura 6.12: Dirección y evolución por hora de la velocidad del viento y la racha

Por último se han incluido las variables precipitación y humedad relativa en un mismo gráfico, debido a la relación entre ambas variables. Se ha optado por un gráfico de barras para la precipitación, debido a que dicha variable puede (y suele) ser 0, con un gráfico de línea superpuesto que indica la humedad relativa de manera horaria. Así, en el ejemplo se puede ver que durante las 19h comenzaron las precipitaciones, que continuaron con menor potencia hasta las 21h, y se aprecia el aumento en la humedad relativa para ese mismo periodo

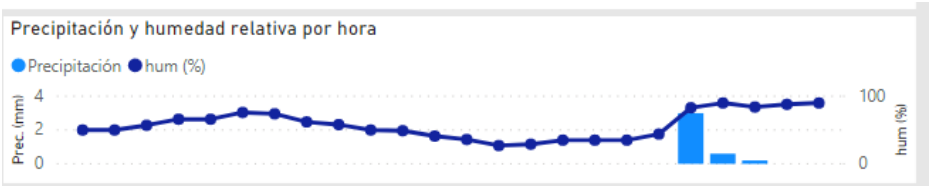


Figura 6.13: Evolución por hora de precipitaciones y humedad

Capítulo 7

Explotación

En este capítulo se materializan los dos último objetivos del proyecto, los de análisis exploratorio y predicción. Para ello se lleva a cabo un Análisis exploratorio de los datos, desde el punto de vista de la influencia de las distintas variables en la presencia de contaminantes. Por último se describe el proceso y resultados de múltiples experimentos llevados a cabo con el objetivo de predecir el volumen horario de cada contaminante.

7.1. Análisis exploratorio

A continuación se realiza un análisis exploratorio de los datos de calidad del aire, desde el 1 de Enero de 2018, hasta los últimos datos registrados en la fecha de realización del análisis (20 de Julio de 2020). Con este análisis buscamos explorar aspectos que nos ayudarán a enfocar correctamente la predicción, con especial hincapié en la relevancia que tienen las variables disponibles en los contaminantes.

Completitud de los datos por estación Comenzamos obteniendo una medida de completitud de los datos por estación. Hay 931 días entre las dos fechas analizadas, por lo que podemos obtener el porcentaje de completitud de datos, sabiendo que para cada día debería haber 24 horas:

nombreEstacion	PorcentajeCompletitud
La rubia II	91,75
Arco de ladrillo II	92,35
Renault1	93
VALLADOLID SUR	93,7
Michelin1	93,76
Renault2	93,89
Puente Regueral	93,92
Renault3	94,26
Vega Sicilia	94,54
Michelin2	94,82
Medina del Campo	96,4

Tabla 7.1: Porcentaje de días con datos por cada estación

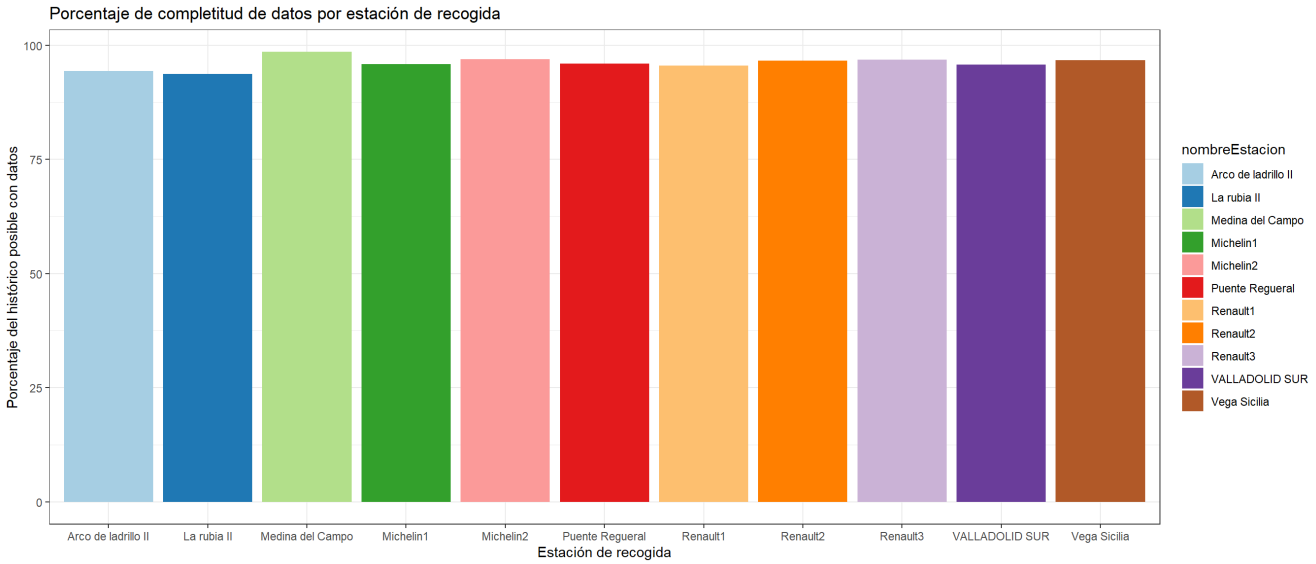


Figura 7.1: Porcentaje de días con datos por cada estación

Todas las estaciones están por encima del 90 % de completitud, con la estación de Medina del Campo siendo la más completa, por encima del 96 % de datos presentes, y La rubia II siendo la menos completa, con alrededor de un 92 % de datos presentes.

Media de horas con datos por estación Otra medida de completitud, importante para el análisis horario de los contaminantes, es la media de horas con datos por día en cada una de las estaciones. Lo deseable es que este dato estuviera lo más cercano a 24 (el día completo) como fuera posible:

nombreEstacion	MediaHorasConDatos	MinHorasConDatos
La rubia II	23,7	1
Michelin1	23,74	2
VALLADOLID SUR	23,75	4
Renault1	23,76	5
Arco de ladrillo II	23,83	7
Puente Regueral	23,83	7
Medina del Campo	23,85	19
Michelin2	23,86	15
Vega Sicilia	23,87	7
Renault2	23,86	6
Renault3	23,92	14

Tabla 7.2: Media y mínimo de horas con datos por día, por estación de medida

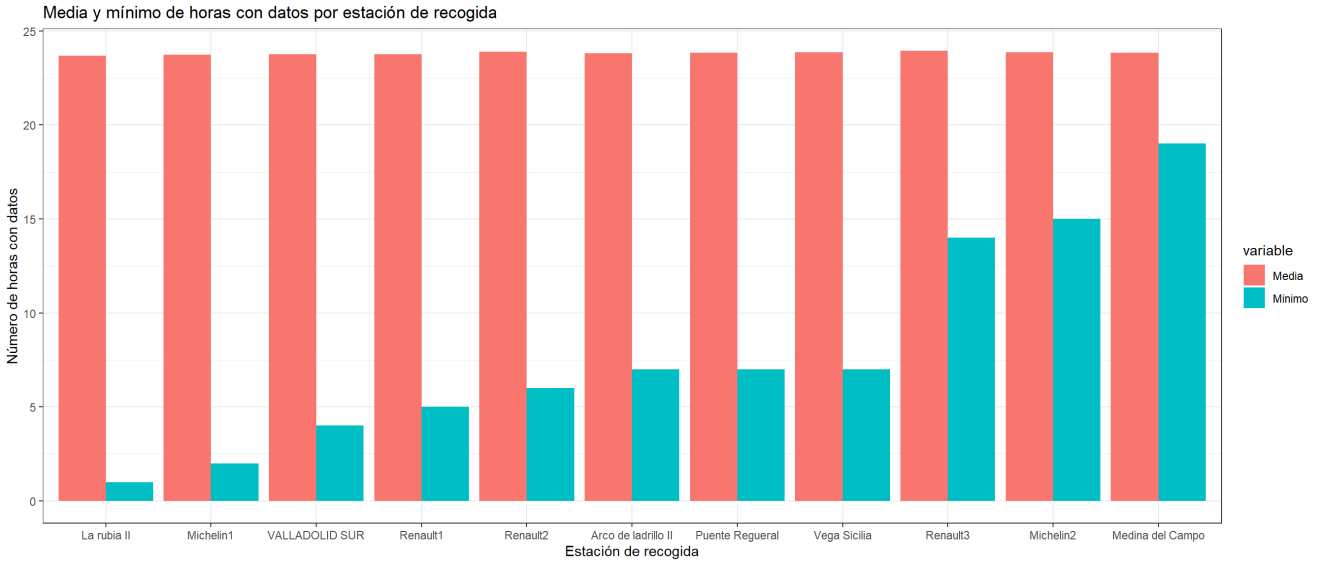
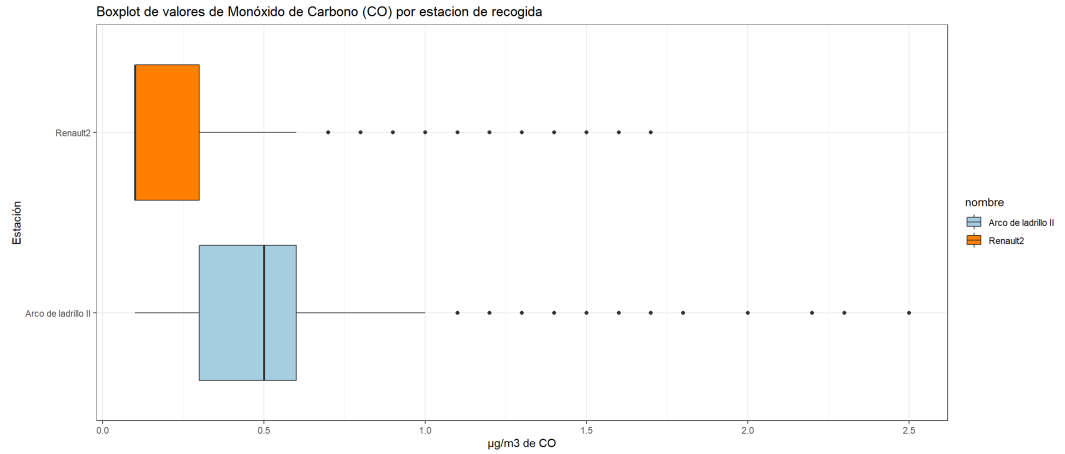


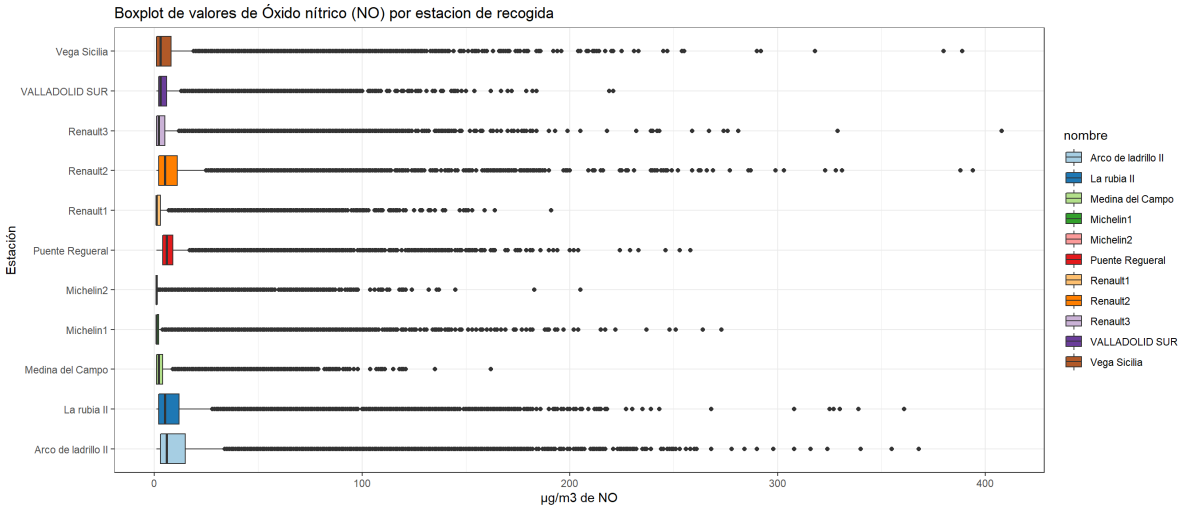
Figura 7.2: Media y mínimo de horas con datos por día, por estación de medida

De media todas las estaciones se aproximan a las 24 horas por día, aunque observando el mínimo de horas, algunas estaciones tienen días con solo 1 hora, como La rubia II, mientras que en otras todos los días son bastante completos, cómo es el caso de Medina del Campo, con 19 horas en su día más incompleto.

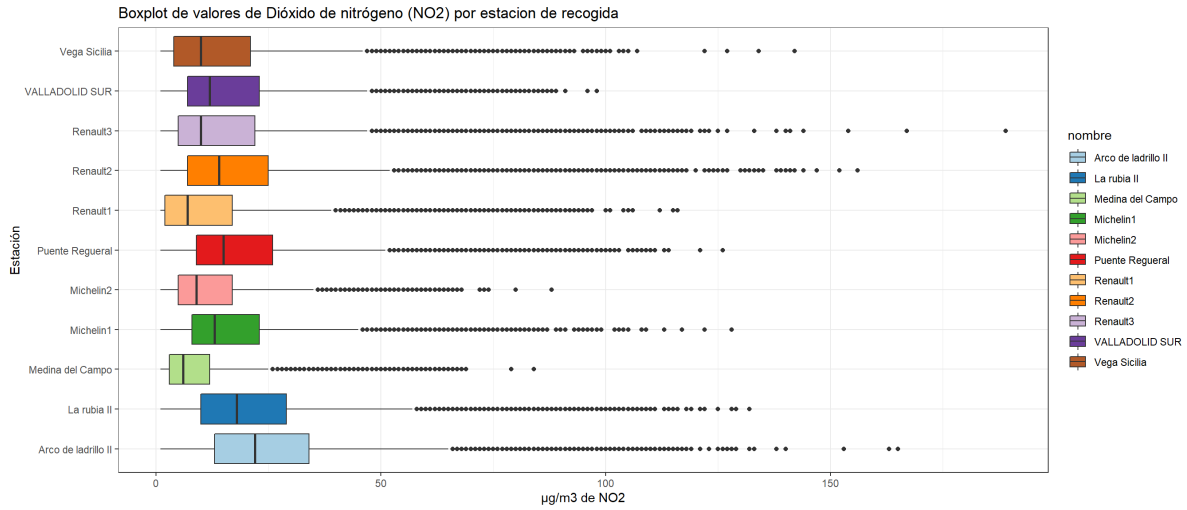
Valores extremos en los datos de contaminantes El primer análisis que se puede hacer de las mediciones de contaminantes es buscar valores extremos que deberán ser considerados cuidadosamente en el análisis. Estos valores pueden ser tanto mediciones incorrectas, como situaciones extremas. Sin embargo, existen algunos ejemplos en los que parece claro que se trata de un error:



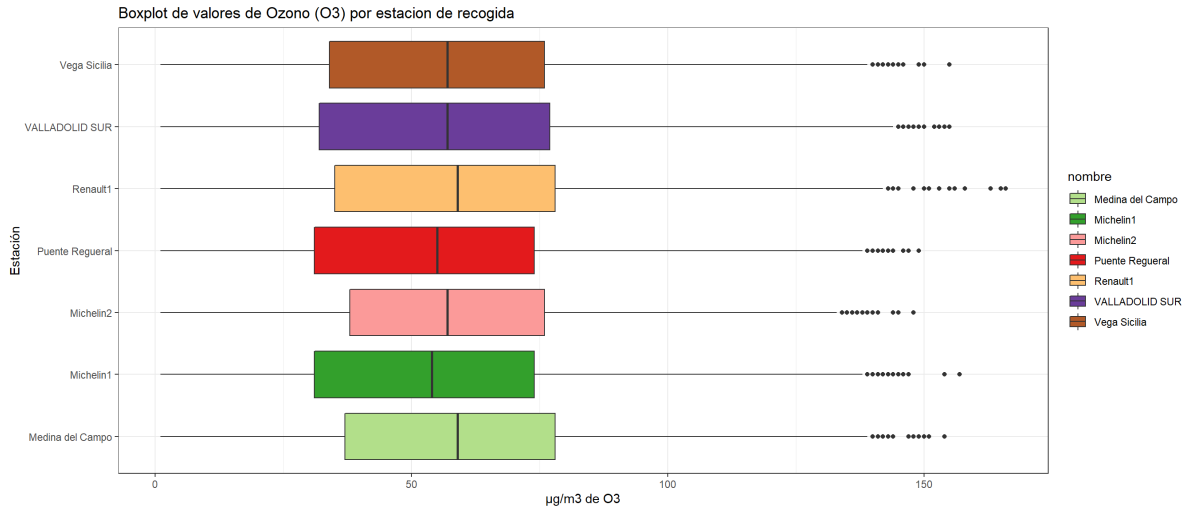
(a) Boxplot de Monóxido de carbono por estación de recogida



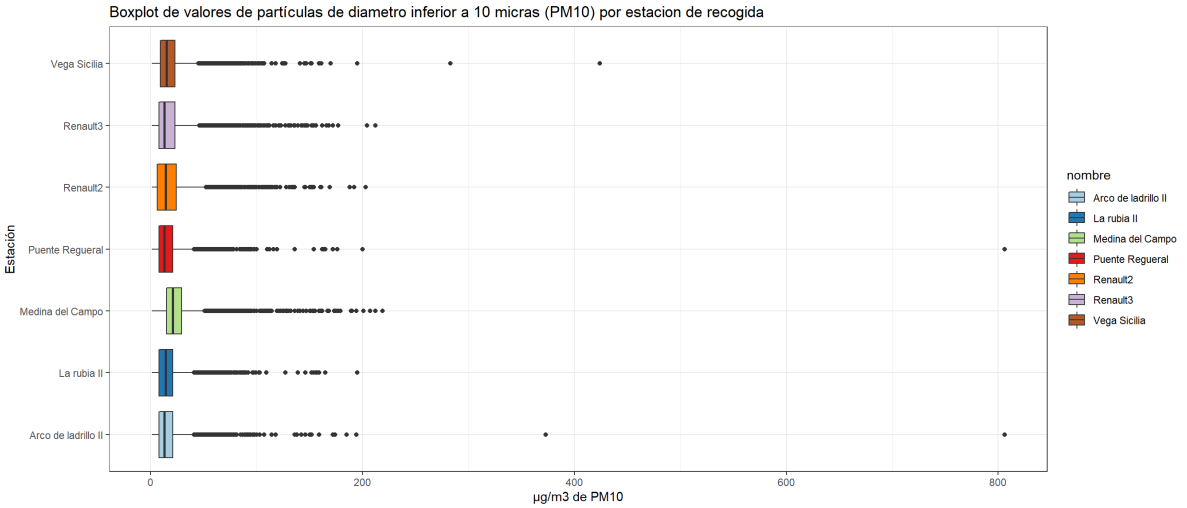
(b) Boxplot de Óxido nítrico por estación de recogida



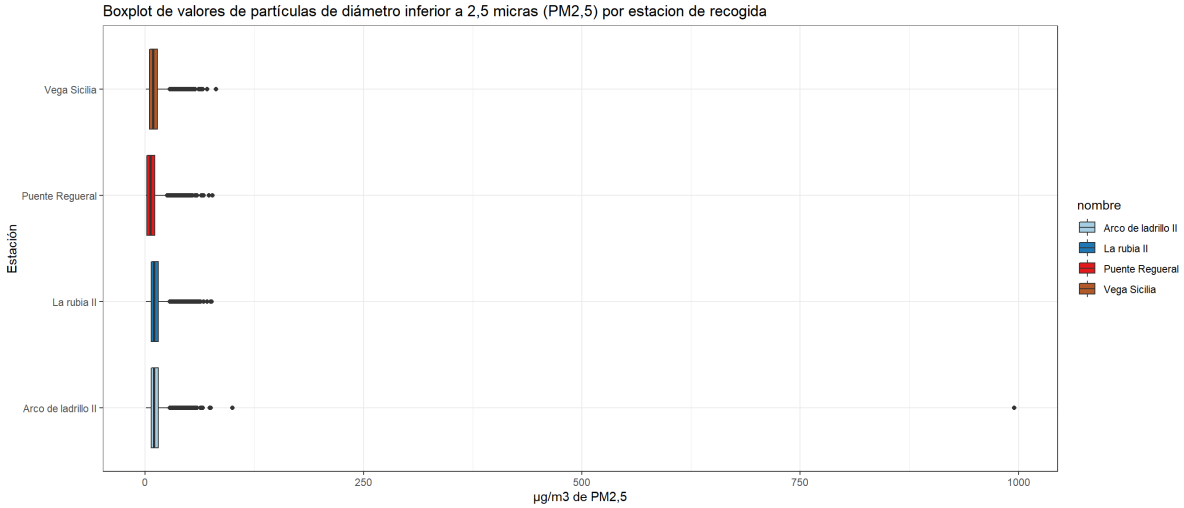
(c) Boxplot de Dióxido de nitrógeno por estación de recogida



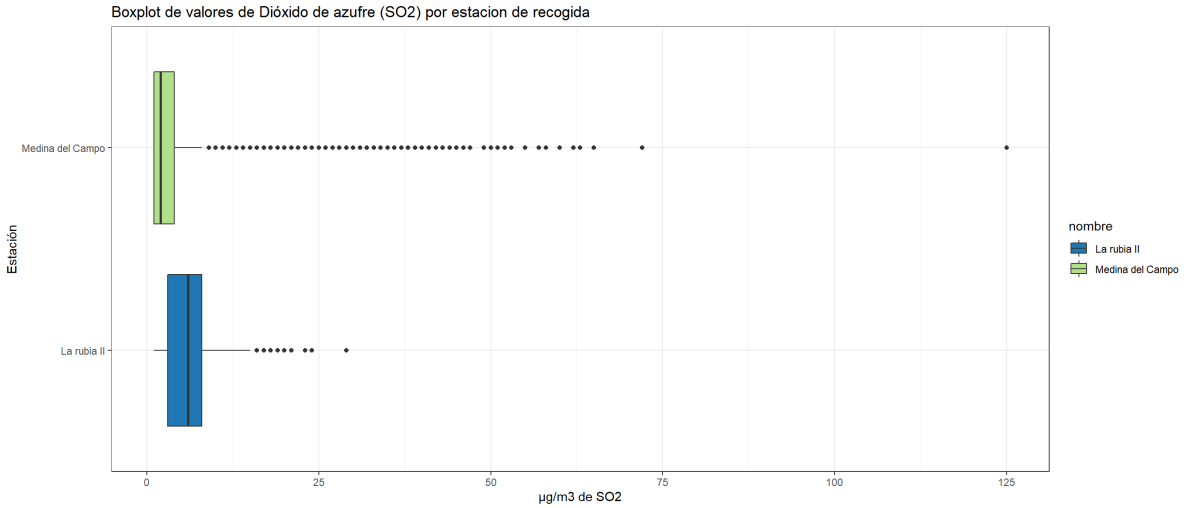
(d) Boxplot de Ozono por estación de recogida



(e) Boxplot de Partículas de diámetro inferior a 10 micras por estación de recogida



(f) Boxplot de Partículas de diámetro inferior a 2,5 micras por estación de recogida



(g) Boxplot de Dióxido de azufre por estación de recogida

Figura 7.3: Contaminantes por estación de recogida

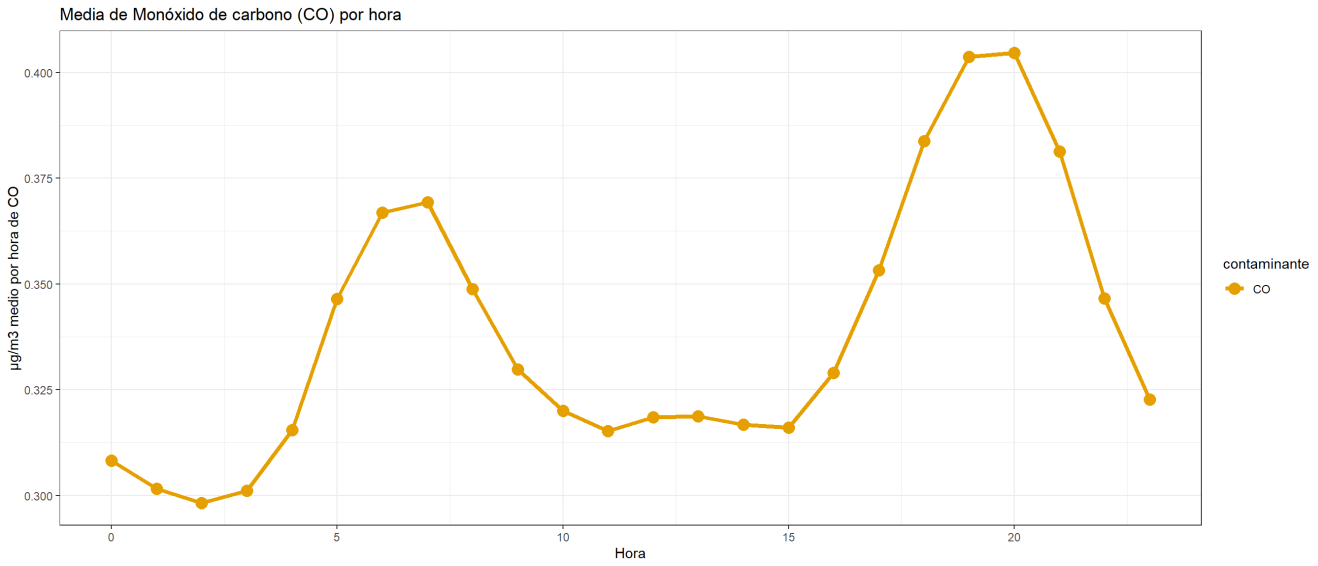
Algunos contaminantes como el CO y el O3 tienen pocos valores extremos, bastante cerca de las distribuciones, con lo que podemos suponer que estos valores son días con contaminante especialmente alto, y no suponen un problema. El NO y el NO2 tienen bastantes valores extremos, los máximos de los cuales se sitúan lejos de las distribuciones, sin embargo por su cantidad podemos esperar que estos valores sean posibles. Por último, ambos

tipos de partículas y el SO2 tienen valores extremos muy separados del resto, y que podemos considerar altamente probable que sean valores de medición. Esto es especialmente claro en el valor de casi 1000 µg/m3 de PM2,5 en Arco de Ladrillo II.

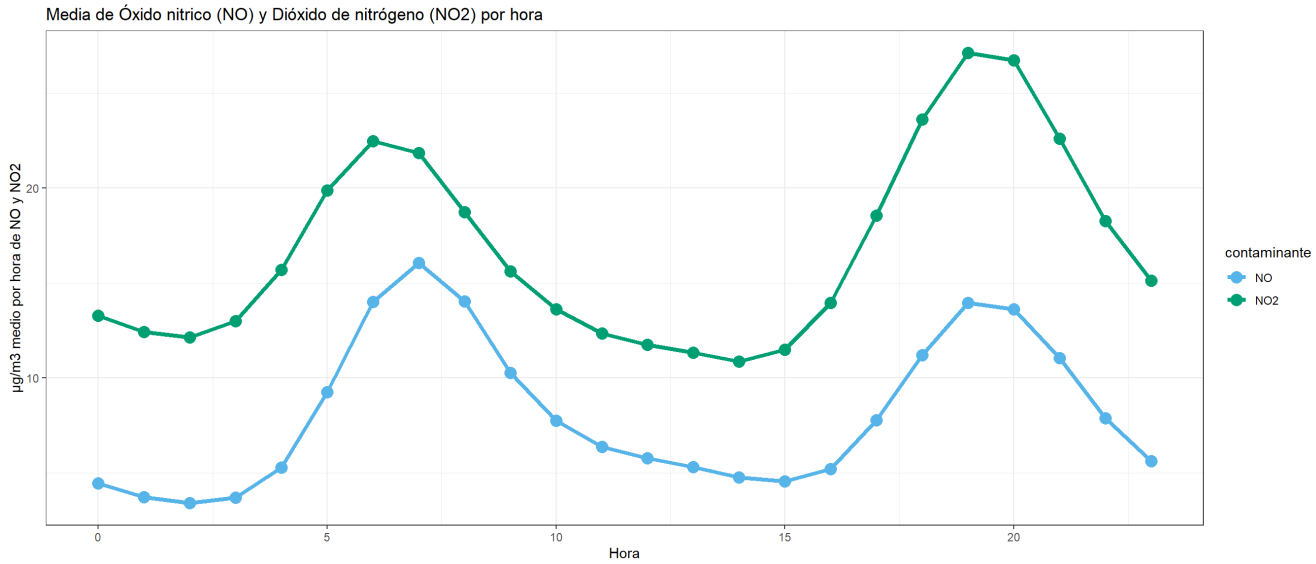
En los siguientes estudios se filtrarán los contaminantes para eliminar estos últimos valores extremos. Específicamente los filtros a aplicar son:

- PM10 $<$ 800 µg/m3
- PM2,5 $<$ 800 µg/m3
- SO2 $<$ 100 µg/m3

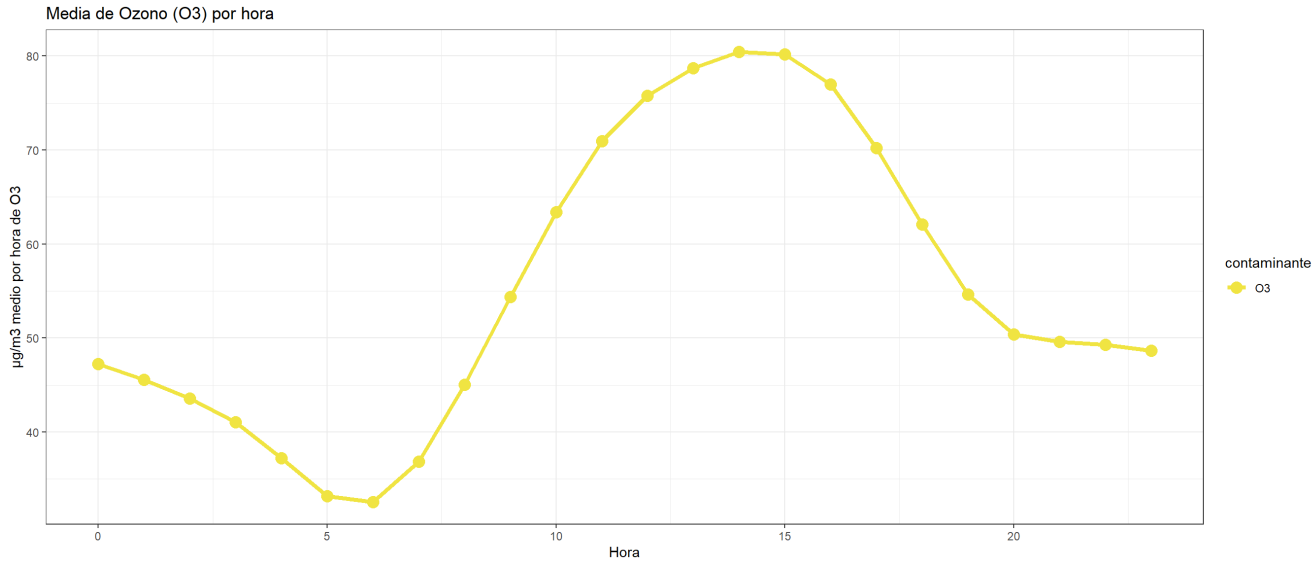
Media horaria de cada contaminante Podemos explorar el patrón horario que experimenta cada contaminante, obteniendo el valor medio de cada contaminante por hora. A continuación se muestran los resultados (algunos contaminantes se han separado al tener escalas de datos muy diferentes al resto):



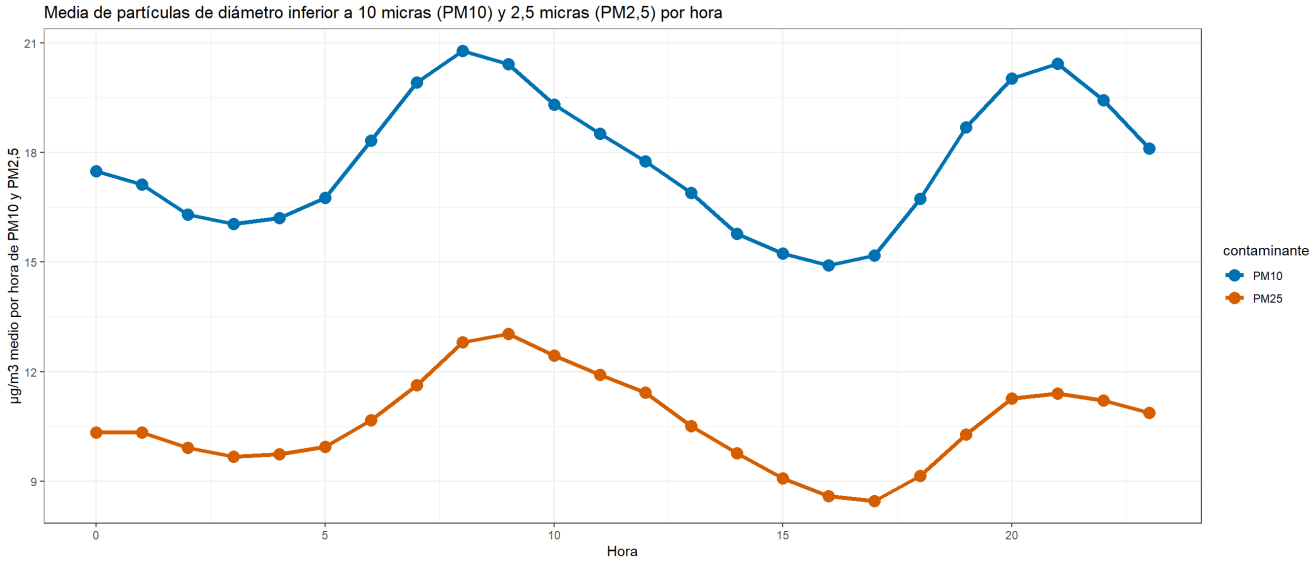
(a) Media por hora de CO



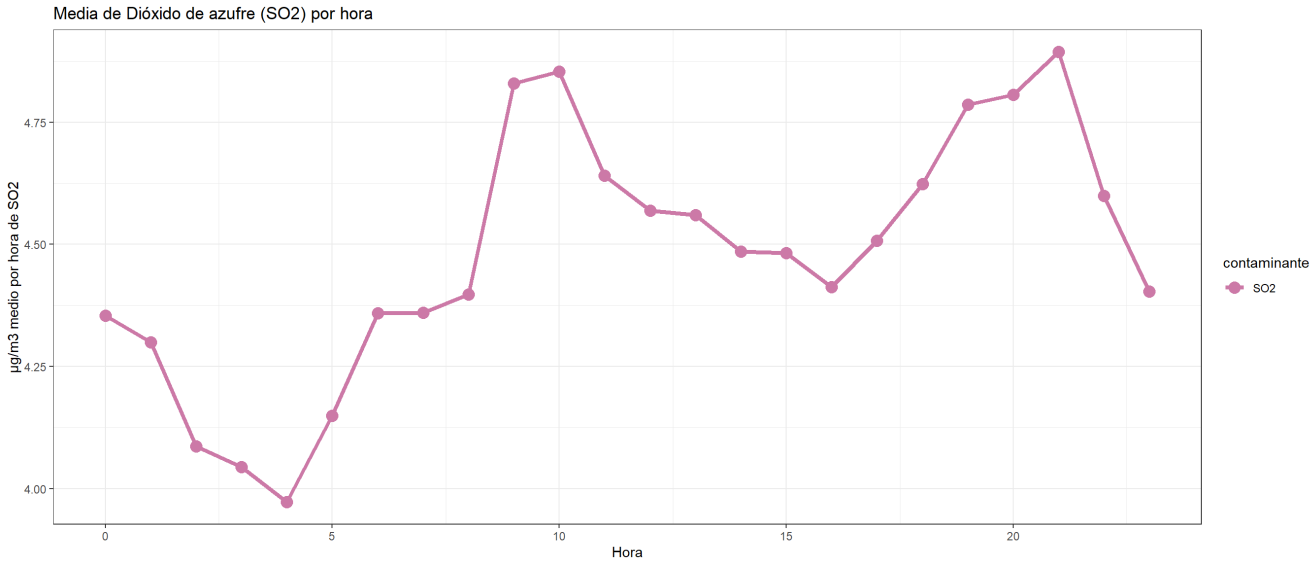
(b) Medias horarias de NO y NO2



(c) Media horaria de O3



(d) Medias horarias de PM10 y PM25



(e) Media horaria de SO2

Figura 7.4: Medias horarias de cada contaminante

La mayoría de contaminantes tienen un patrón claro en su media horaria. Específicamente el CO y el NO2 tiene dos picos, el primero alrededor de las 6 y las 7 de la mañana, y más tarde un pico aún mayor y más largo, con comienzo a las 6 de la tarde y finalización sobre las 8 de la noche. Esto se podría corresponder con horarios laborales, en los que hay un mayor movimiento de vehículos, que producen dichos contaminantes.

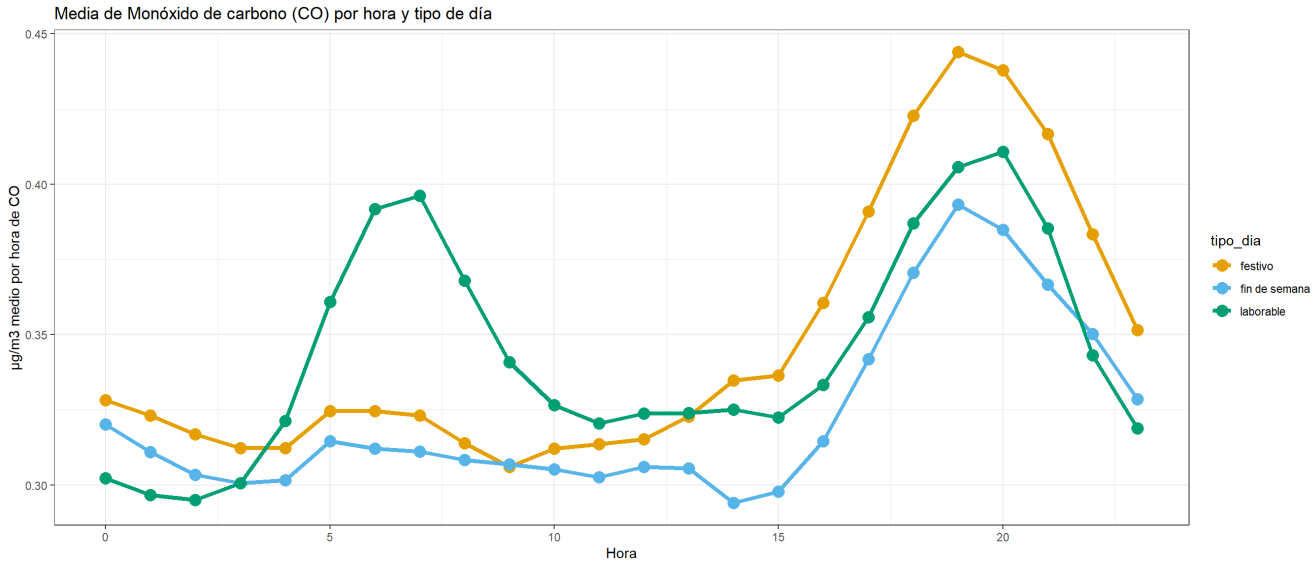
Los contaminantes NO y partículas PM10 y PM25 tienen un patrón similar al anterior, solo que el segundo pico

no llega a superar al de la mañana.

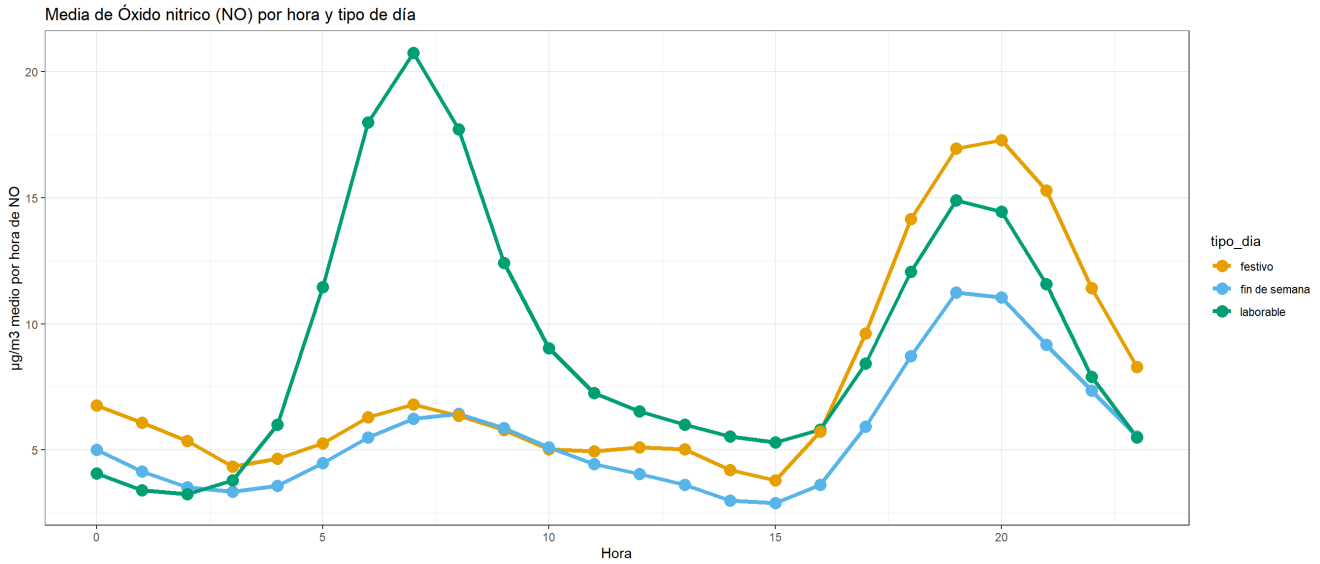
El ozono tiene un comportamiento distinto, comenzando a aumentar a partir de las 7 de la mañana, alcanzando su pico a las 2 de la tarde, y cayendo hasta las 8 de la tarde, a partir de cuando desacelera su caída hasta alcanzar el mínimo a las 5 de la mañana. Esto es consecuente con su origen, ya que se produce por combinaciones de varios elementos reaccionando a la luz del sol, por lo que es de esperar que sus valores más altos se produzcan durante el día, cayendo durante la noche, en ausencia de la luz solar.

El SO2 no parece mantener un patrón reconocible, más allá de bajar durante la noche y tener valores más altos durante el día.

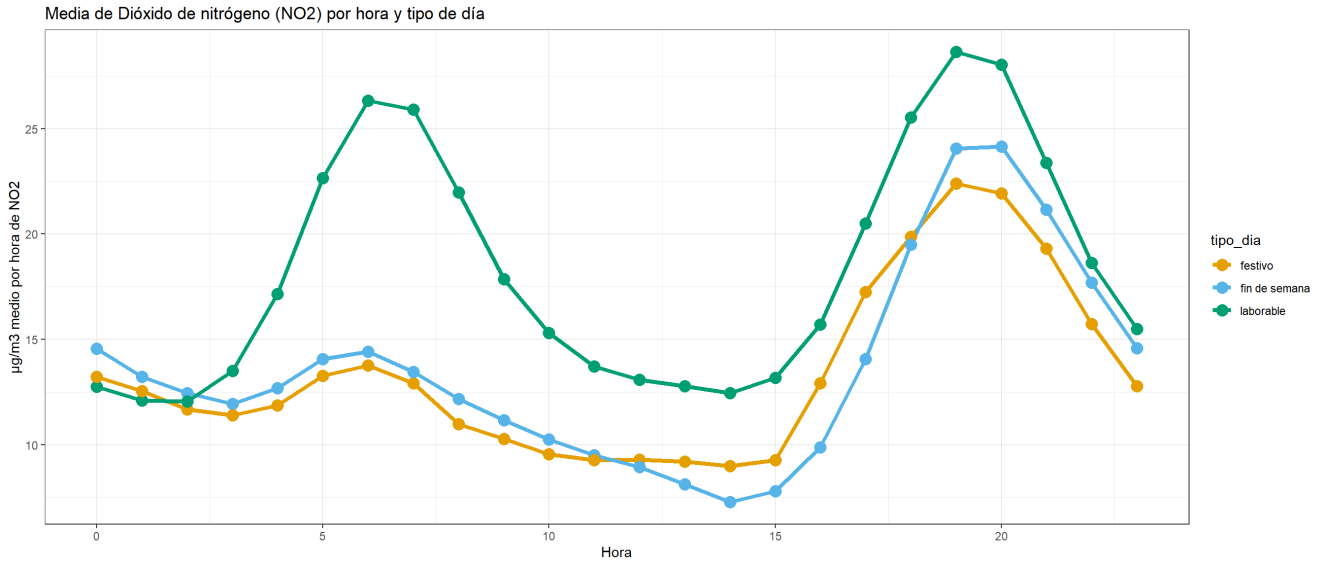
Efecto del tipo de día sobre la media horaria de contaminantes Relacionado con el estudio anterior, y como confirmación de que algunos de estos patrones están relacionados con actividades laborables, podemos observar la diferencia en dichos patrones según si el día fue festivo en Valladolid, fin de semana (Sábado o Domingo) o de Lunes a Viernes laborable.



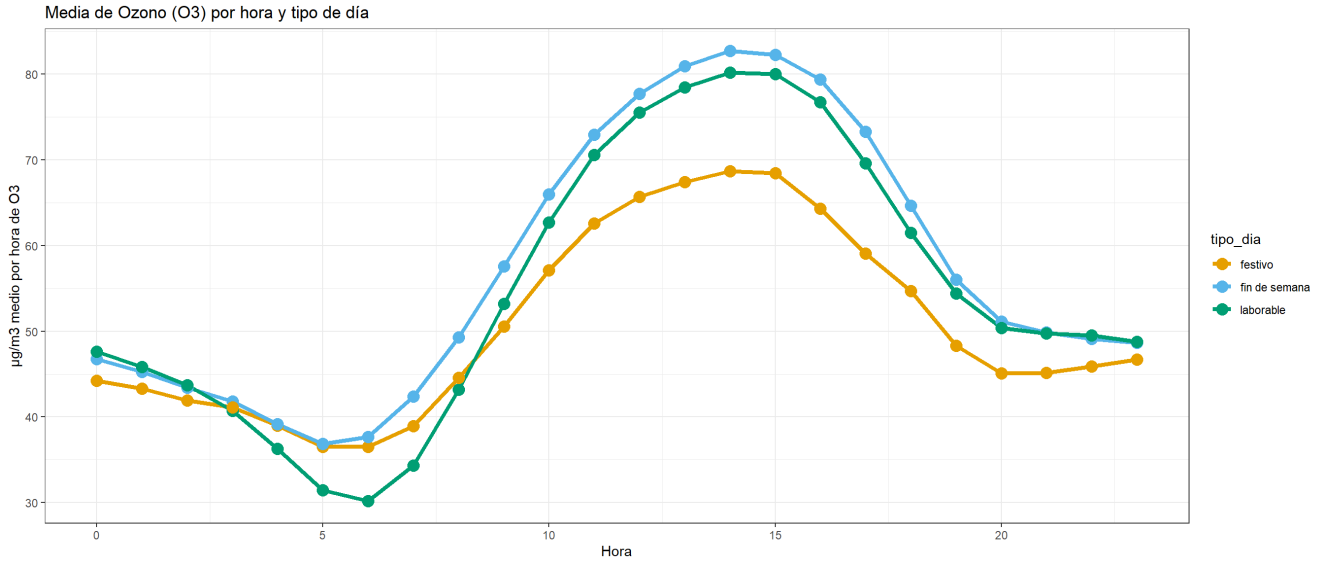
(a) Medias horarias de CO en días festivos y no festivos



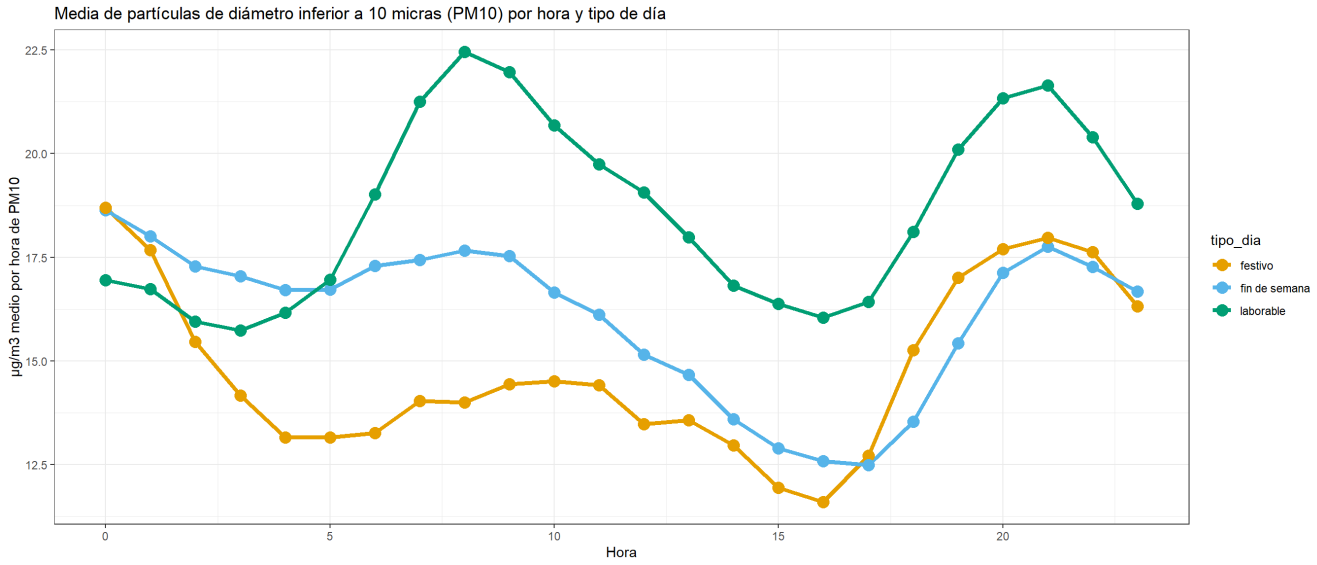
(b) Medias horarias de NO en días festivos y no festivos



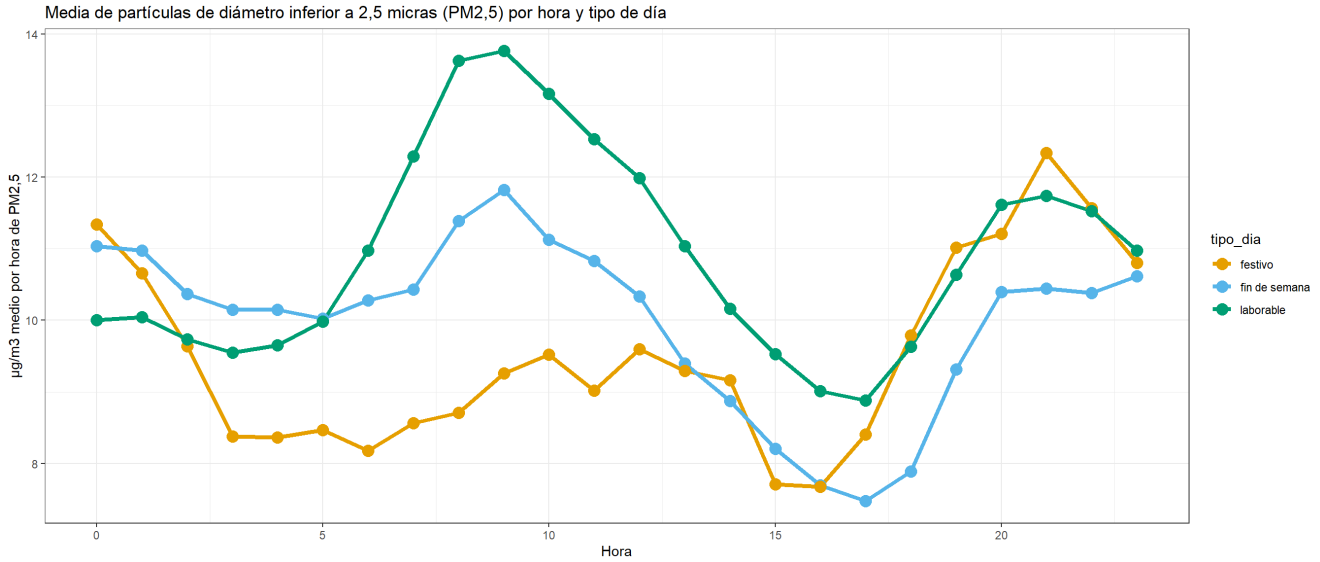
(c) Medias horarias de NO2 en días festivos y no festivos



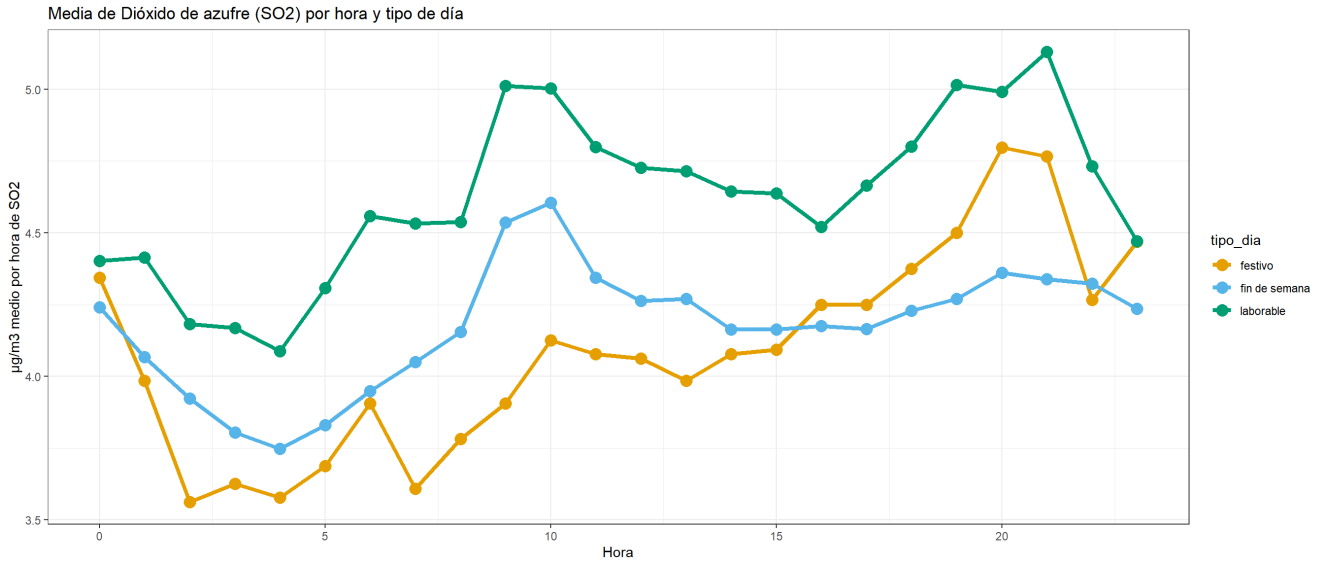
(d) Medias horarias de O3 en días festivos y no festivos



(e) Medias horarias de PM10 en días festivos y no festivos



(f) Medias horarias de PM25 en días festivos y no festivos



(g) Medias horarias de SO2 en días festivos y no festivos

Figura 7.5: Medias horarias de cada contaminante separado en días festivos y no festivos

Los patrones de CO y NO se ven modificados de la misma manera: el primer pico desaparece cuando consideramos fines de semana y festivos, mientras que el segundo pico aumenta en los festivos por encima de los laborables y fines de semana.

El NO2 pierde también el primer pico en los días festivos y fines de semana, pero a diferencia de los anteriores, los valores durante festivos y fines de semana se sitúan por debajo del dato para días laborales durante el segundo

pico.

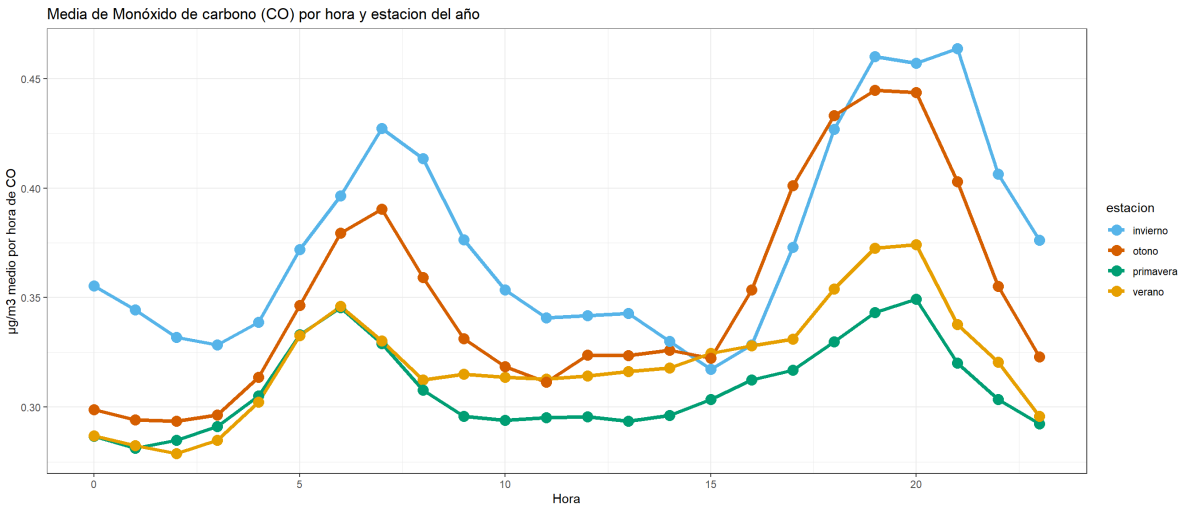
El O3 experimenta un patrón muy similar en los tres tipos de día, con festivos teniendo un valor significativamente menor, y fines de semana teniendo valores un poco más altos que los días laborables.

Las partículas PM10 experimentan un primer pico en los festivos a las 12 de la noche, con una caída sucesiva para situarse por debajo del valor de días laborables para el resto de horas. En general los días laborables tienen valores más altos excepto previo a las 5 de la mañana, donde en los fines de semana se produce de media más contaminante.

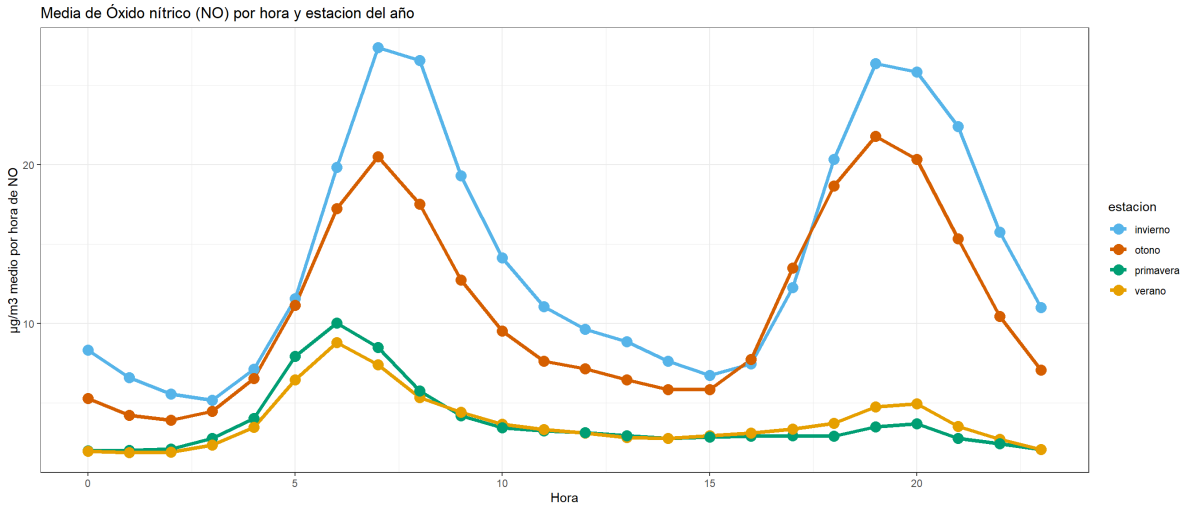
El otro tipo de partículas tiene también un pequeño pico a las 12 de la noche los festivos, con otro valor similar a las 9 de la noche. El resto del comportamiento es similar al de PM10.

Por último, el SO2 tiene patrones similares en laborables y fines de semana, mientras que los festivos son bajos hasta las 2 de la tarde, cuando empiezan a aumentar.

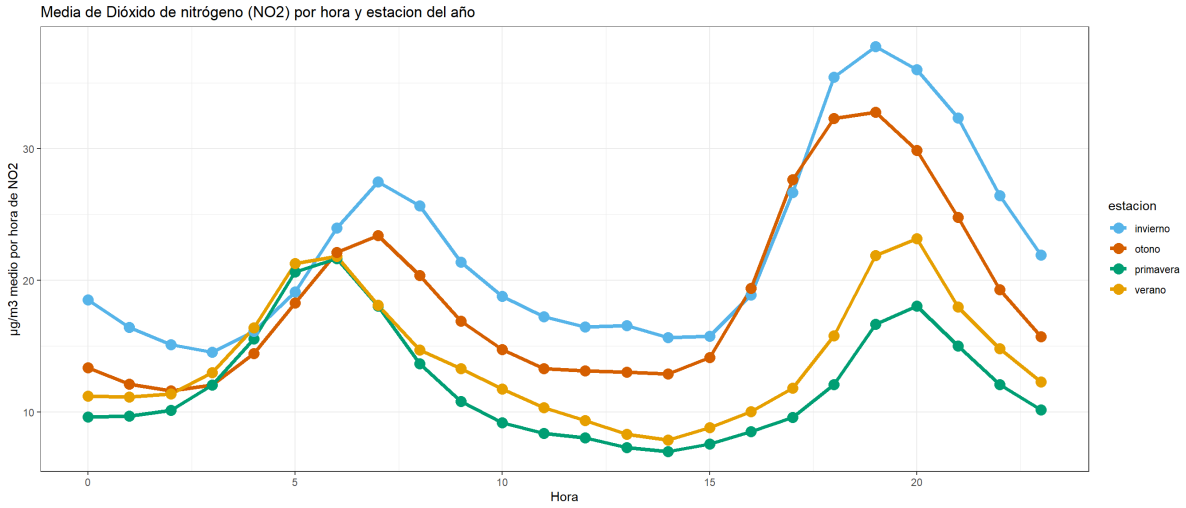
Media horaria para cada contaminante por estación del año Queremos también detectar si hay diferencias significativas en los patrones o valores de los contaminantes según la estación del año.



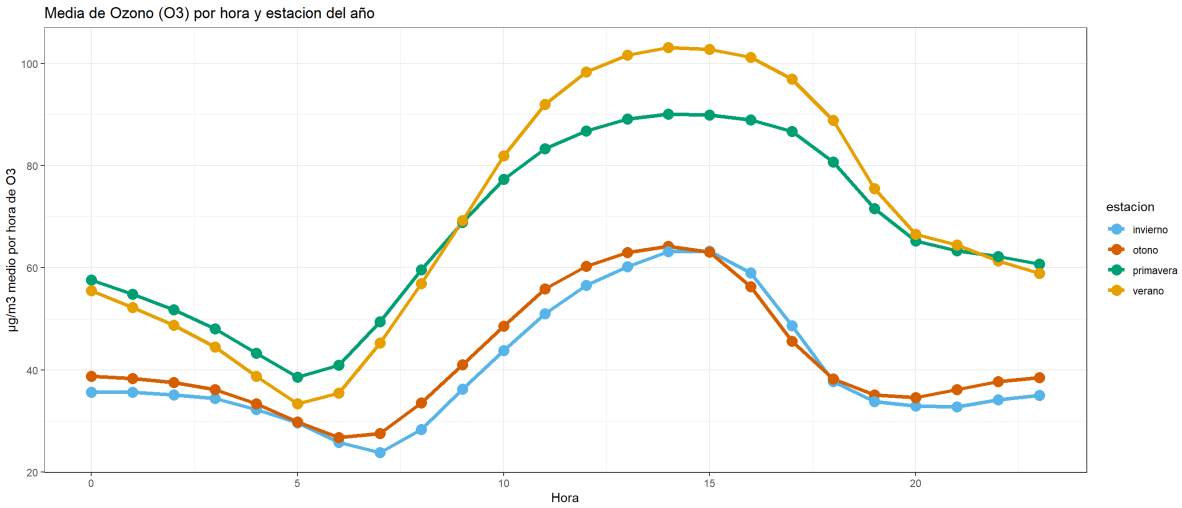
(a) Medias horarias de CO por estación del año



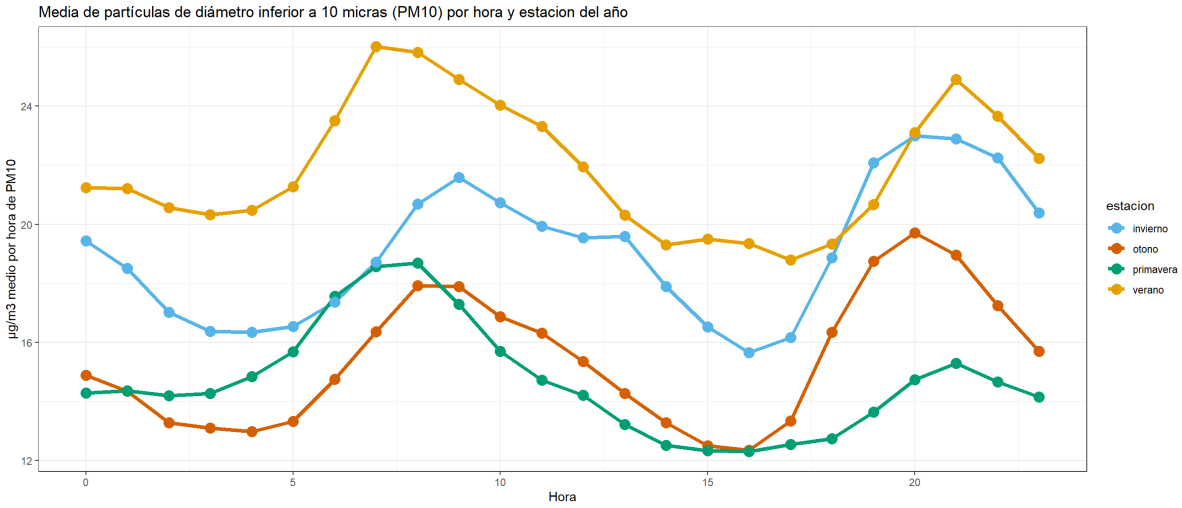
(b) Medias horarias de NO por estación del año



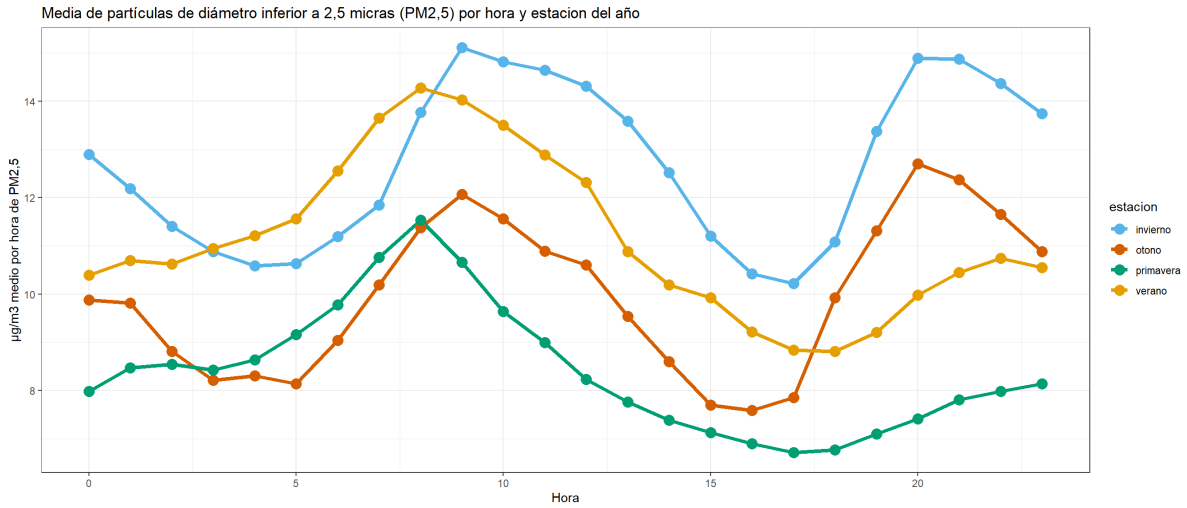
(c) Medias horarias de NO2 por estación del año



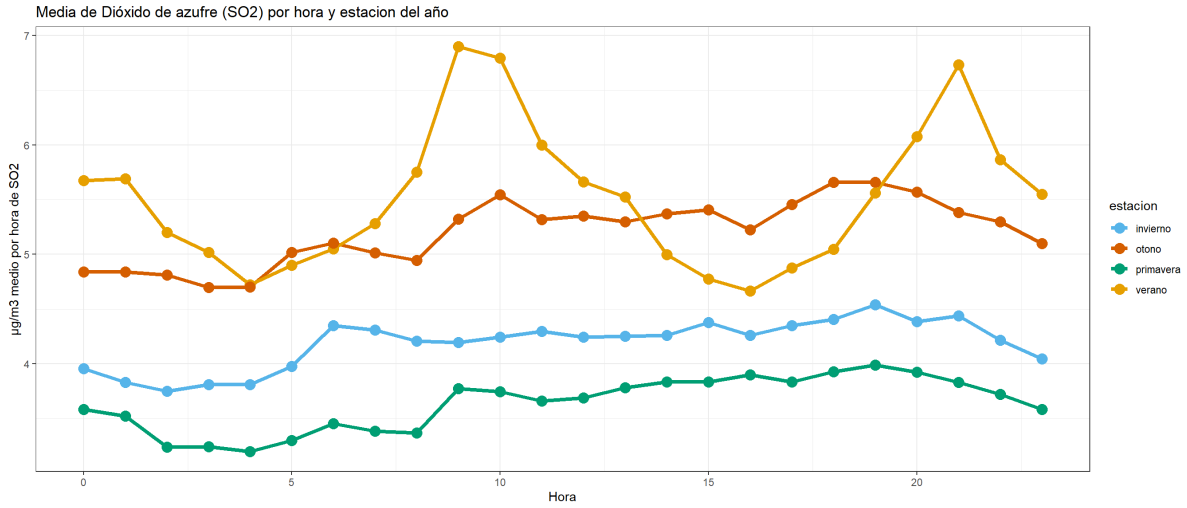
(d) Medias horarias de O3 por estación del año



(e) Medias horarias de PM10 por estación del año



(f) Medias horarias de PM25 por estación del año



(g) Medias horarias de SO2 por estación del año

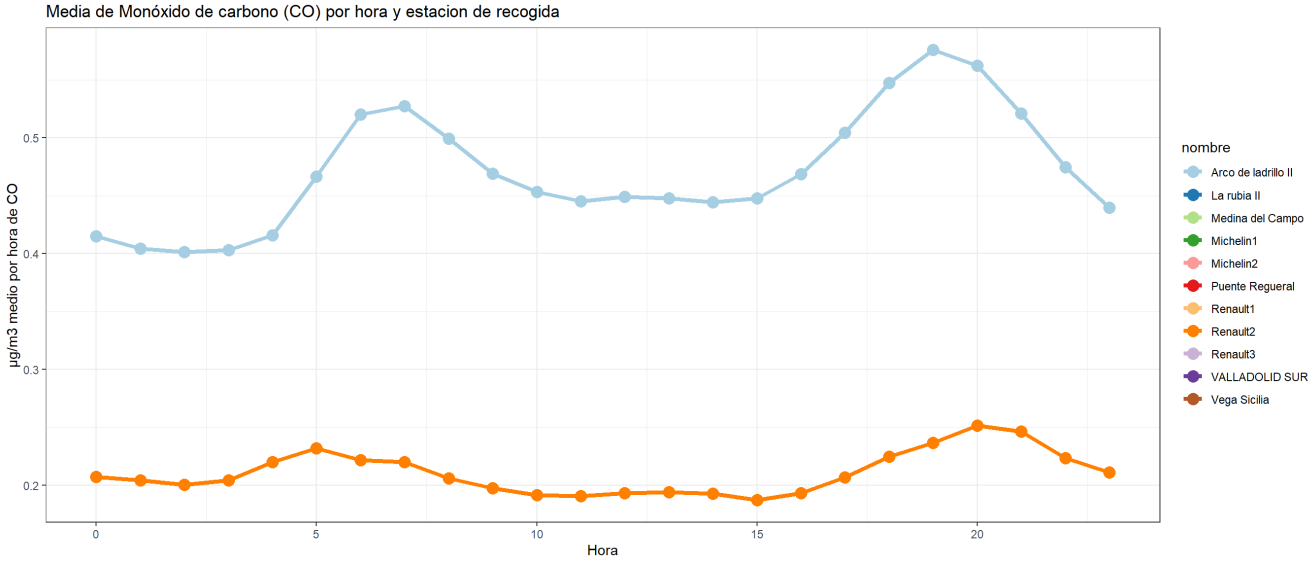
Figura 7.6: Medias horarias de cada contaminante por estación del año

Se aprecian diferencias en la media horaria entre estaciones, y se pueden detectar cuatro grupos de contaminantes:

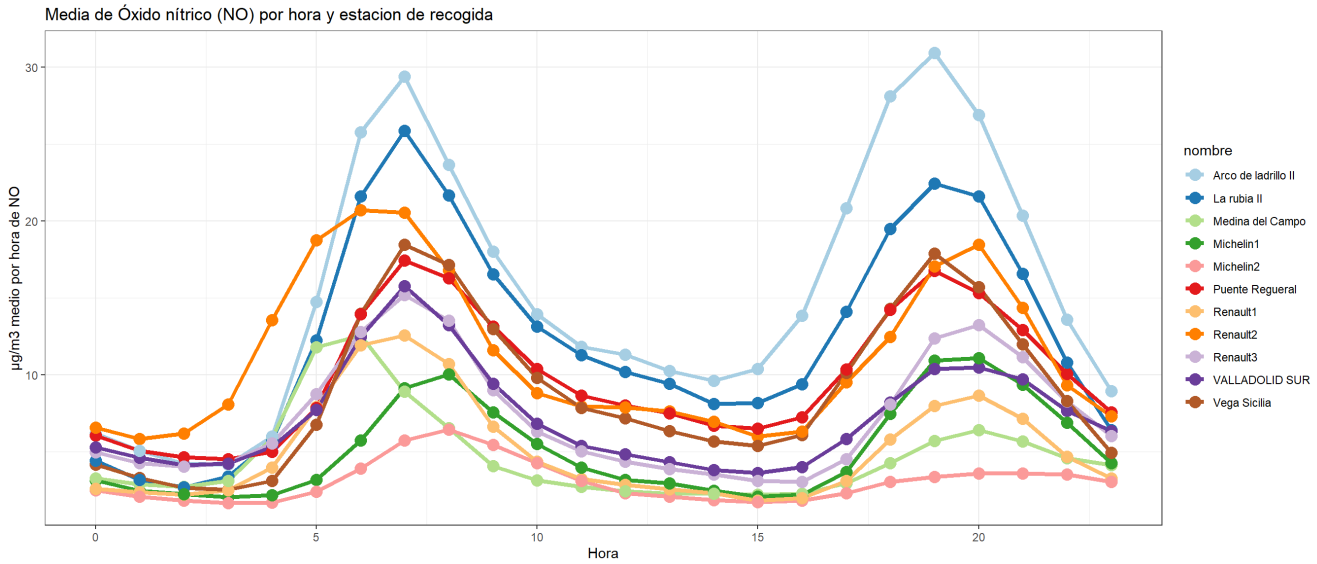
- Aquellos que tienen valores más altos durante las estaciones más frías (invierno y otoño), lo cual puede tener relación con calefacciones y mayor uso de vehículos a motor: CO, NO y NO2.
- O3, que alcanza valores más altos en estaciones cálidas (verano y primavera), relacionado con un mayor número de horas de luz solar, y menores en invierno y otoño, ya que el NO aumenta en estas estaciones, y el O3 disminuye con la presencia de NO.

- PM10 y PM25, que alcanzan valores mayores durante el invierno y el verano, también aumentando durante la madrugada en primavera y las tardes y noches de otoño.
- Por último el SO2, con valores más altos en otoño y verano, aunque esta última estación tiene una caída significativa durante las tardes.

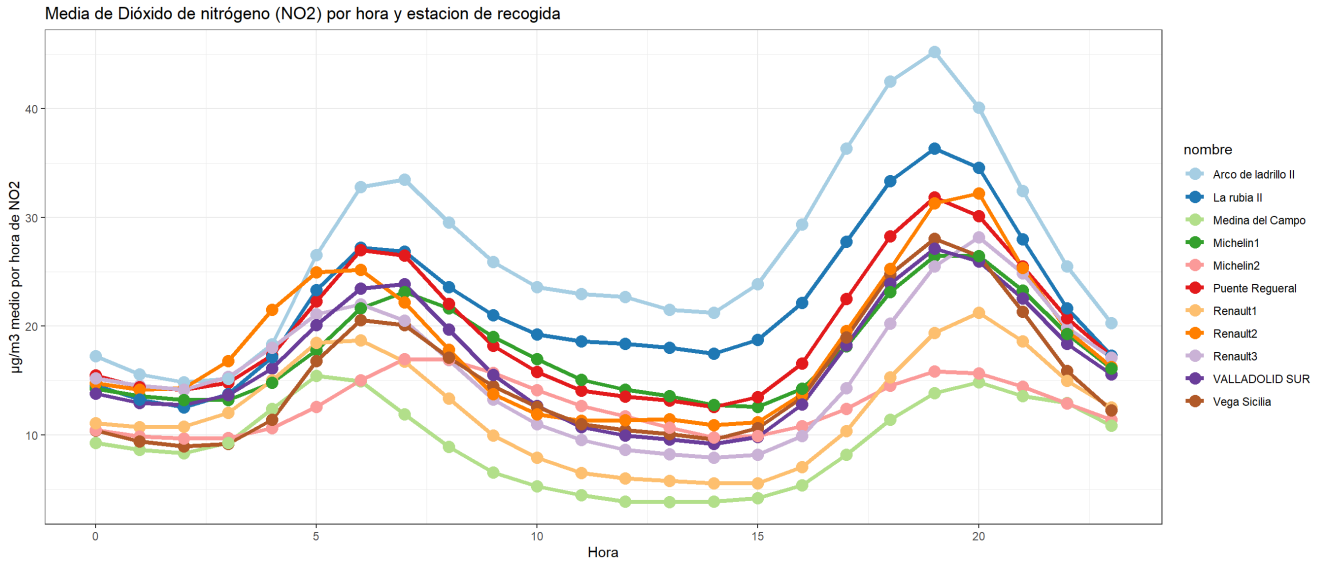
Media horaria para cada contaminante por estación de recogida Otro factor que puede ser de interés es cómo afecta la estación de recogida a la media horaria de contaminantes. Las estaciones de recogida de datos están repartidas en distintos puntos de la ciudad de Valladolid con características diversas (por ejemplo, la estación de Arco de Ladrillo II se sitúa en el interior de una zona verde como es el Campo Grande, mientras que las estaciones Renault1 y Renault2 se sitúan cerca de las fábricas de Renault en la N-601) , y por tanto se espera que el valor de contaminantes dependa significativamente de la estación que las recoge.



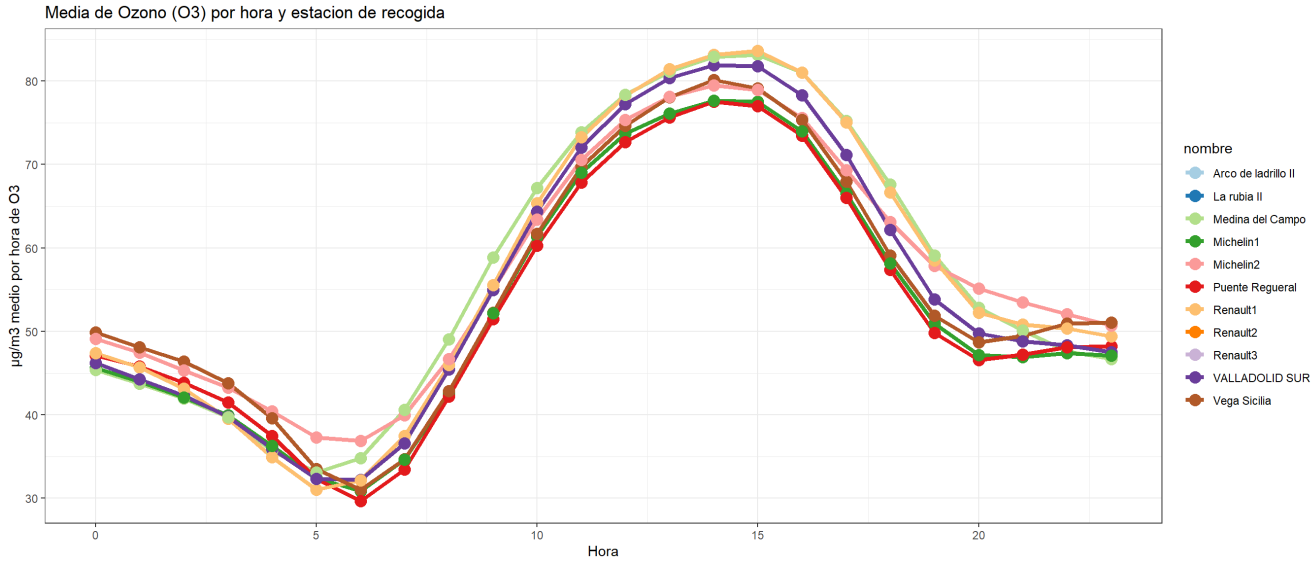
(a) Medias horarias de CO por estación de recogida



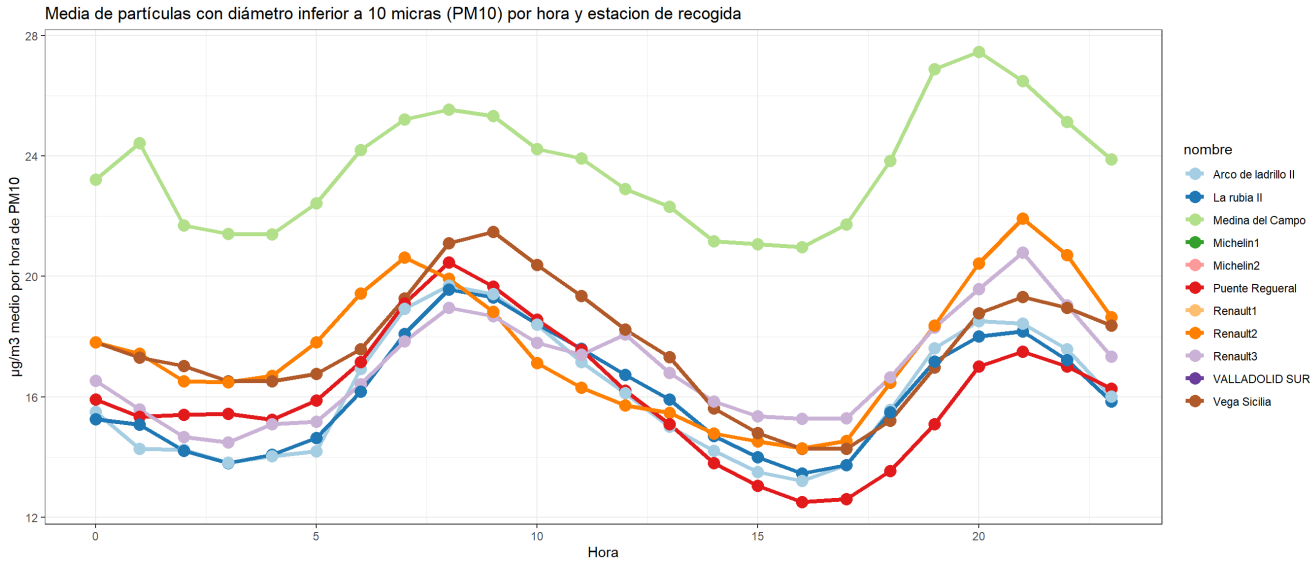
(b) Medias horarias de NO por estación de recogida



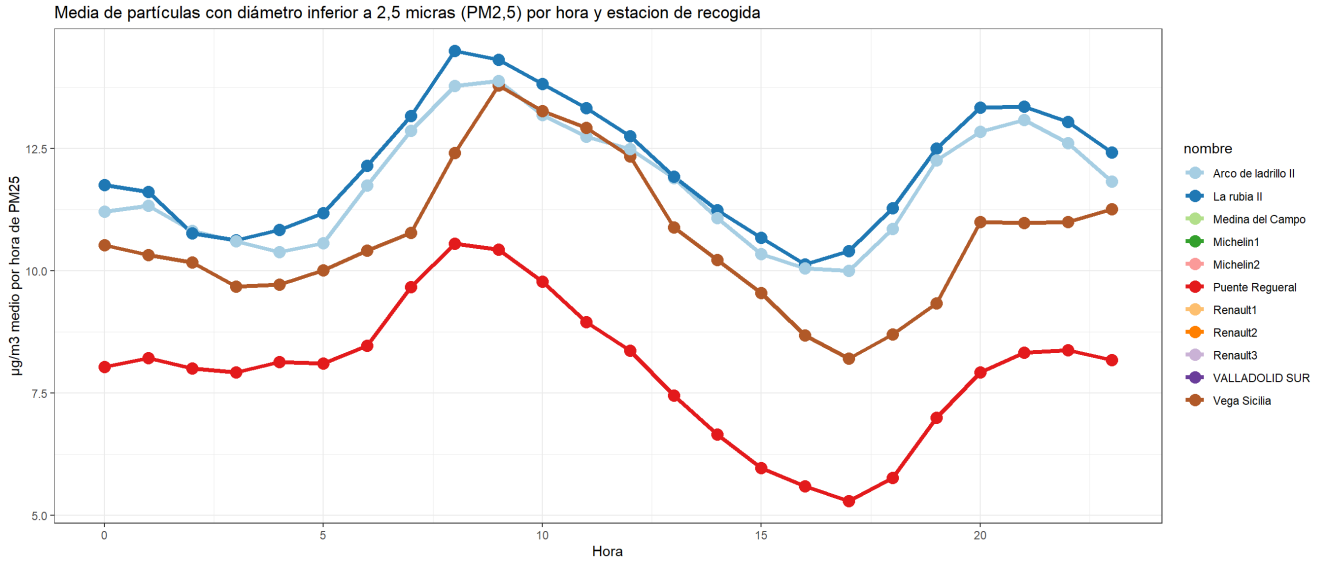
(c) Medias horarias de NO2 por estación de recogida



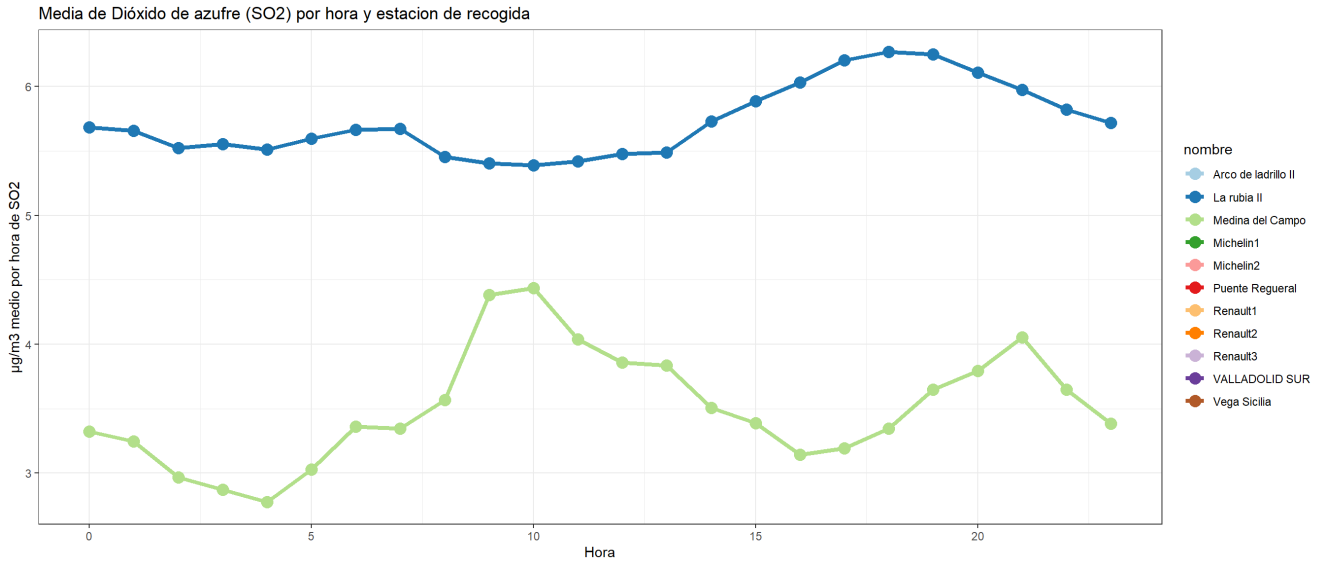
(d) Medias horarias de O3 por estación de recogida



(e) Medias horarias de PM10 por estación de recogida



(f) Medias horarias de PM25 por estación de recogida



(g) Medias horarias de SO2 por estación de recogida

Figura 7.7: Medias horarias de cada contaminante por estación de recogida

Se pueden observar diferencias significativas en algunos contaminantes, como CO o SO2, mientras que algunos, como el O3, no varían demasiado con la estación de medida. En el caso del CO, NO y NO2, Arco de Ladrillo II tiene valores más altos que el resto de estaciones, aunque en NO se puede observar que el primer pico de valores para la estación Renault2 se adelanta alrededor de 1 hora del pico del resto de estaciones, indicando quizá que las actividades que producen NO cerca de dicha estación (posiblemente actividad de la fábrica y vehículos a motor) tienen una hora de comienzo distinta a las que lo producen en otras.

En el caso de partículas PM10, la estación de Medina del Campo tiene valores significativamente más altos que el resto de estaciones, mientras que los valores de SO2 en La rubia II superan por bastante a los de la otra estación que lo mide (Medina del Campo).

Completitud de los datos meteorológicos Como sabemos, un factor importante para la calidad del aire es el estado meteorológico de la zona. Por ello es de importancia la completitud de los datos meteorológicos de los que disponemos. También se ha indicado anteriormente que, debido a la naturaleza de la fuente de recogida de datos, sólo se tienen datos de meteorología desde Noviembre de 2019. El resultado es un porcentaje de completitud del 24 %.

El porcentaje de días y horas de histórico con datos parcialmente completos de meteorología (contienen al menos una de las variables meteorológicas disponibles) es del 24,47 %

Se puede también calcular el porcentaje de completitud de cada variable meteorológica:

variable	PorcentajeCompletitud
Temperatura	23,2
Velocidad viento	23,18
Dirección viento	22,72
Velocidad racha	23,18
Direccion racha	23,18
Precipitación	23,19
Presión	23,2
Humedad	23,2

Tabla 7.3: Porcentaje de días y horas con datos por variable de meteorología

Dirección del viento es la variable menos completa, mientras que el resto parece tener porcentajes de completitud muy similares.

Relación entre variables numéricas Podemos trazar una correlación entre todas las variables de contaminantes y meteorología, generando una matriz de correlaciones de Spearman (no lineales) entre los valores horarios. Para ello consideramos solo las observaciones de cada pareja de variables donde ambas variables están completas (por ejemplo esto generará una falta de correlación entre CO y O3, ya que ninguna estación mide tanto CO como O3).

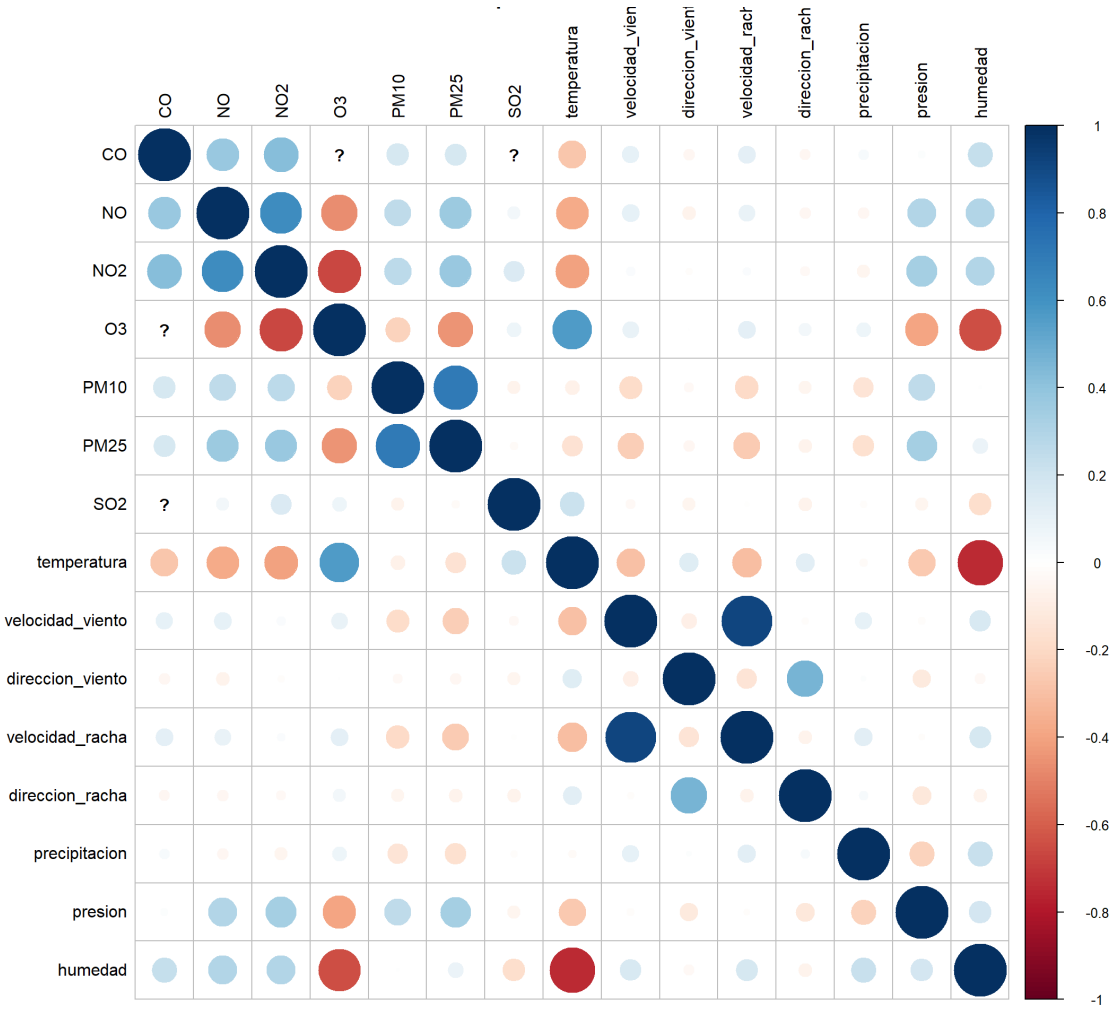


Figura 7.8: Matriz de correlaciones entre las variables numéricas

El tamaño de los círculos indica la fuerza de la relación, mientras que el color muestra la dirección (azul indica relación positiva, ambas variables aumentan juntas, y rojo indica relación negativa, cuando una variable aumenta la otra disminuye).

El ozono es la variable con más correlaciones fuertes, específicamente relaciones negativas con NO2 y humedad, y positivas con la temperatura. Todas estas relaciones se explican debido a cómo se consume y produce el O3, ya que el O3 se consume rápidamente en presencia de NO, para producir NO2. La temperatura y la humedad juegan un papel clave en la producción de O3, ya que este se produce en presencia de luz solar. La temperatura es una variable relacionada positivamente con la insolación (cuanta mayor fuerza de la luz solar, más altas serán las temperaturas), mientras que la humedad es una variable indicadora de cielos nublados, con la consecuente reducción de luz solar.

Otras relaciones de importancia son las producidas entre las velocidades de viento y racha, que aumentan juntas, así como la temperatura y la humedad, que están inversamente correlacionadas.

Las variables de NO y NO2 están positivamente relacionadas, ya que las fuentes de estos contaminantes son las mismas. Si embargo, esta relación no es demasiado fuerte, probablemente por la oxidación del NO para producir NO2 mencionada anteriormente, cuya relación tiene un carácter claramente negativo.

Las partículas PM10 y PM2,5 están correlacionadas positivamente, probablemente debido a que sus fuentes son similares. No se aprecia una relación significativa entre PM2,5 y NO2, SO2, temperatura o humedad, cuyas reacciones son en teoría orígenes secundarios de PM2,5.

Conclusiones Con este análisis hemos conseguido observar valores extremos probablemente incorrectos para algunos contaminantes, útil en el caso de querer utilizar algoritmos de predicción sensibles a valores extremos. También hemos afianzado la confianza en que la mayoría de contaminantes siguen patrones horarios, y que existen ciertas variables categóricas que podemos calcular (estación del año y tipo de día), y que son valiosas para la predicción, ya que algunos contaminantes son muy sensibles a estas.

Respecto a las variables meteorológicas, la mayoría se ven influenciadas en mayor o menor medida por la temperatura, la presión y la humedad, excepto por el SO2, que no parece tener ninguna influencia entre las variables numéricas de las que se dispone. Los contaminantes también están correlacionados.

7.2. Análisis predictivo

Como se ha indicado en las secciones 2.3.1 y 3.3, el análisis predictivo llevado a cabo evalúa tres algoritmos distintos para cada contaminante, eligiendo sus parámetros utilizando validación cruzada de cinco particiones. Para ello se lleva a cabo el proceso detallado en el siguiente esquema, que se repite para cada contaminante:

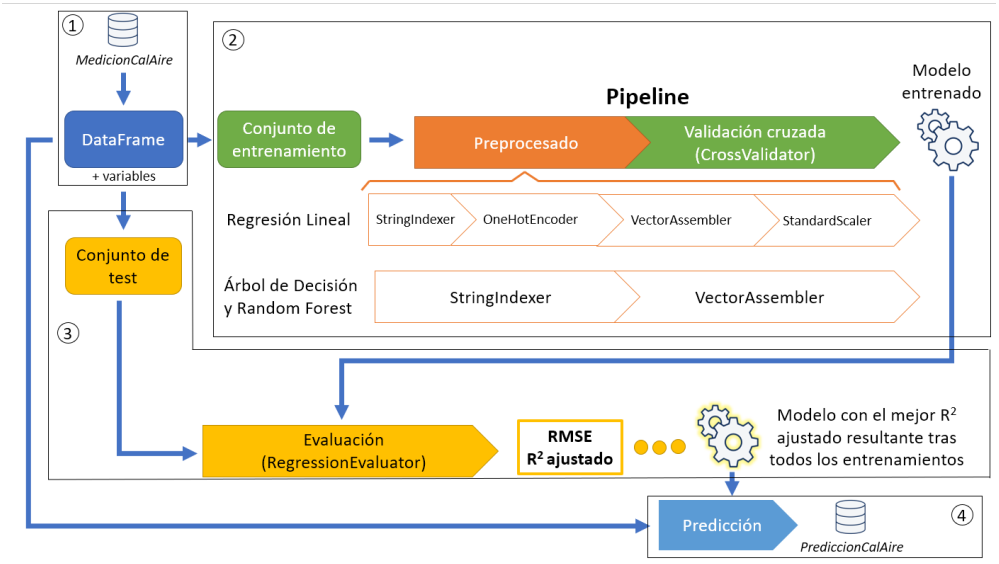


Figura 7.9: Esquema del proceso de predicción de contaminantes

1. Los datos se encuentran en la base de datos TFM dentro del SQL Server, específicamente se utiliza la tabla MedicionCalAire. Esta se lee y transforma a un DataFrame (estructura de datos de Spark ML) mediante un controlador oficial de Microsoft[1]. A este DataFrame se le añaden tres variables con las que queremos representar algunos de los factores que hemos detectado que pueden tener relevancia a la hora de predecir algunos contaminantes. Específicamente creamos las variables *tipoDia*, *estacionAño* y *franjaHoraria*, esta última representando franjas de 5 horas (Madrugada (0-4), mañana (5-9), mediodía (10-14), tarde (15-19), noche (20-23)).

2. Se seleccionan dos tercios de las instancias aleatoriamente como conjunto de entrenamiento y validación. A continuación se introducen los datos a un pipeline. Este objeto de Spark ML permite ejecutar varios procesos secuencialmente, en nuestro caso el contenido del pipeline dependerá de si se trata de un experimento de Regresión Lineal o no. A continuación se describe el significado de cada etapa posible del Pipeline:

   - *StringIndexer:* Este proceso transforma las variables categóricas en un índice numérico. Por ejemplo, una variable con valores [primavera, verano, otoño, invierno] pasaría a contener los valores [0, 1, 2, 3].
   - *OneHotEncoder*: Mientras que el proceso anterior convierte variables categóricas en números, es una representación que no podemos utilizar para la predicción, ya que expresa que, por ejemplo, la estación invierno es mayor que la estación primavera ($3 > 1$) y a su vez, que hay tanta diferencia entre el verano y el otoño, como entre el otoño y el invierno (ambos pares de estaciones estarían una posición la una de la otra). Para evitar que el modelo parta de suposiciones como estas, se convierte cada variable en un vector de *n-1* posiciones (siendo n el número de categorías en la variable) que contendrá 1 en el elemento correspondiente a la categoría y 0 en el resto. En nuestro ejemplo, el valor primavera sería (0,0,0), verano (0,0,1), otoño (0,1,0) e invierno (1,0,0). Este proceso no se ejecuta en el árbol de decisión y random forest ya que estos algoritmos aceptan variables categóricas tal y como son, y detectaría simplemente una variable con cuatro categorías sin asumir nada sobre su orden o valor numérico.
   - *VectorAssembler:* Este proceso toma todas las variables a introducir al modelo y las convierte en un Vector, que es la estructura de datos que utilizan como entrada los modelos de Spark ML.
   - *StandardScaler*: La última etapa del pre-procesado, estandariza las variables regresoras, convirtiendo cada una en una variable con media 0 y desviación estándar 1. Esto es necesario cuando tenemos variables de entrada con distintas unidades, ya que ciertos algoritmos son sensibles a diferencias en magnitudes de las variables de entrada. Esto no es necesario para árboles de decisión o random forest, ya que no consideran diferencias entre magnitudes de variables para su aprendizaje.
   - *CrossValidator:* Esta etapa es la que lleva a cabo la Validación Cruzada explicada en la sección 2.3.1, tiene como salida el modelo con los mejores parámetros encontrados, entrenado y listo para su evaluación.

3. Una vez entrenado un modelo, se evalúa utilizando el conjunto de test, lo que nos da una medida de cómo de eficaz es el modelo en predecir datos nunca vistos. Esto se lleva a cabo con la clase *RegressionEvaluator*, que toma el valor real y la predicción de un modelo y da como salida una serie de métricas. Los pasos 2 y 3 se repiten para los tres algoritmos y, en el caso de la Regresión Lineal, para múltiples combinaciones de variables, tras lo cual nos quedamos con el modelo que muestra el mejor R^2 ajustado.

[1] Disponible en https://www.microsoft.com/es-es/download/confirmation.aspx?id=54671. Visitado por última vez el 18 de Septiembre de 2020

4. Por último utilizamos el mejor modelo resultante de los entrenamientos para predecir el contaminante para el conjunto de datos completo, y lo almacenamos en la tabla *PrediccionCalAire*.

Spark ML permite guardar en un fichero cualquier modelo, tras lo cual podemos leer y utilizarlo sin tener que volver a pasar por el entrenamiento. Esto nos permitiría realizar predicciones de datos nuevos de manera rápida tras haber entrenado los modelos.

Para entrenamiento y test de los modelos se han utilizado los datos a partir del 16 de Noviembre de 2019, ya que son aquellos para los que se tienen datos de meteorología, con el objetivo de que todos los modelos entrenados tengan aproximadamente el mismo tamaño de dataset inicial. Respecto al tamaño del dataset, este variará según el contaminante, ya que algunos contaminantes son medidos por más estaciones de recogida que otros, por tanto teniendo mayor número de observaciones. En cualquier caso, se toman 247 días de datos, con carácter horario, lo que implica aproximadamente 6000 observaciones horarias, que en cada caso son multiplicadas por el número de estaciones de recogida midiendo dicho contaminante.

Para que los experimentos se puedan repetir y se obtengan los mismos resultados, se ha utilizado una semilla, lo que permite convertir las operaciones aleatorias que suele conllevar un modelo de aprendizaje automático, en operaciones deterministas.

7.2.1. Selección y evaluación del modelo

Los parámetros entrenados en el caso de las regresiones lineales son elasticNetParam y regParam entre valores [0 0,1 0,3 0,5 0,7 1]. Para el árbol de decisión se entrena el parámetro maxDepth entre [5 7 9 11 13 15 17 23 27 30]. Para el random forest, el parámetro numTrees entre [5 10 20 40 80 160 320].

O3 - Regresión lineal Se ha seguido una metodología mediante la cual se prueba a introducir nuevas variables al modelo, manteniéndolas siempre que mejoren el R^2ajustado de manera significativa (en más de 0,001). Si la variable no mejora la métrica o la empeora, no pasa a formar parte del modelo siguiente.

Por ejemplo, primero se prueba el modelo O3 = estacionRecogida, y se obtiene un valor para la métrica. Si la métrica mejora con el modelo O3 = estacionRecogida + temperatura, este es el nuevo modelo al que se añaden las siguientes variables. Si no mejora, se continúa con el modelo O3 = estacionRecogida.

El tiempo de procesado de los modelos ronda los 5 minutos hasta el modelo 9, a partir del cual comienzan a subir los tiempos de procesado, aunque siempre por debajo de los 10 minutos.

	Nueva variable probada	RMSE (menos mejor)	R^2 adj (más mejor)	Parámetros óptimos	Mejora significativa
1	estacionRecogida	29,878	0,002	elasticNetParam=0,0 regParam=0,1	
2	temperatura	20,56	0,423	elasticNetParam=0,0 regParam=0,0	SI
3	presión	19,366	0,48	elasticNetParam=0,0 regParam=0,0	SI
4	humedad	17,727	0,563	elasticNetParam=1,0 regParam=1,0	SI
5	velocidad_viento	17,481	0,591	elasticNetParam=1,0 regParam=1,0	SI
6	velocidad_racha	17,308	0,597	elasticNetParam=1,0 regParam=1,0	SI
7	direccion_viento	17,235	0,593	elasticNetParam=0,0 regParam=0,0	NO
8	direccion_racha	17,31	0,597	elasticNetParam=1,0 regParam=1,0	NO
9	estacionAño	16,752	0,623	elasticNetParam=0,0 regParam=0,0	SI
10	tipoDia	16,675	0,626	elasticNetParam=0,0 regParam=0,0	SI
11	franjaHoraria	16,022	0,654	elasticNetParam=1,0 regParam=1,0	SI

Tabla 7.4: Resultados de los experimentos de Regresión Lineal para la predicción de Ozono

Modelo final: O3 = estacionRecogida + temperatura + presión + humedad + velocidad_viento + velocidad_racha + estacionAño + tipoDia + franjaHoraria

O3 - Árbol de decisión En este caso no es necesario probar múltiples modelos, ya que el propio árbol de decisión elige sólo las mejores variables. El algoritmo tarda alrededor de 11 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 10,433
- **R^2ajustado:** 0,847
- **Parámetro óptimo**: maxDepth=23

O3 - Random Forest Análogo al árbol de decisión. El algoritmo tarda alrededor de 9 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 15,718

- **R^2ajustado:** 0,653
- **Parámetro óptimo**: numTrees=5

NO - Regresión lineal El tiempo de procesado de los modelos ronda los 3 minutos hasta el modelo 8, con un tiempo de procesado alrededor de 10 minutos, hasta el modelo 11 que tiene un tiempo de procesado de 50 minutos.

	Nueva variable probada	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	temperatura	15,806	0,068	elasticNetParam=0,0 regParam=0,0	
2	presión	15,871	0,11	elasticNetParam=1,0 regParam=1,0	SI
3	humedad	15,916	0,109	elasticNetParam=1,0 regParam=1,0	NO
4	velocidad_viento	15,577	0,108	elasticNetParam=1,0 regParam=1,0	NO
5	velocidad_racha	15,177	0,12	elasticNetParam=1,0 regParam=1,0	SI
6	direccion_viento	13,75	0,113	elasticNetParam=1,0 regParam=1,0	NO
7	direccion_racha	15,224	0,125	elasticNetParam=1,0 regParam=1,0	SI
8	estacionAño	14,963	0,154	elasticNetParam=1,0 regParam=1,0	SI
9	tipoDia	14,844	0,167	elasticNetParam=1,0 regParam=1,0	SI
10	franjaHoraria	14,552	0,199	elasticNetParam=1,0 regParam=1,0	SI
11	estacionRecogida	14,374	0,205	elasticNetParam=0,0 regParam=0,0	SI

Tabla 7.5: Resultados de los experimentos de Regresión Lineal para la predicción de Monóxido de Nitrógeno

Modelo final: NO = temperatura + presión + velocidad _racha + direccion_racha + estacionAño + tipoDia + franjaHoraria + estacionRecogida

NO - Árbol de decisión El algoritmo tarda alrededor de 11 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 7,533
- **R^2ajustado:** 0,742
- **Parámetro óptimo**: maxDepth=5

NO - Random Forest El algoritmo tarda alrededor de 14 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 12,693
- **R^2ajustado:** 0,2679
- **Parámetro óptimo**: numTrees=5

NO2 - Regresión lineal El tiempo de procesado de los modelos ronda los 4 minutos hasta la inclusión de variables categóricas en el modelo 8, cuando aumentan los tiempos de procesado hasta aproximadamente10 minutos.

	Nueva variable probada	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	temperatura	12,788	0,093	elasticNetParam=0,0 regParam=0,0	
2	presión	12,352	0,166	elasticNetParam=1,0 regParam=1,0	SI
3	humedad	12,388	0,165	elasticNetParam=1,0 regParam=1,0	NO
4	velocidad_viento	12,414	0,156	elasticNetParam=1,0 regParam=1,0	NO
5	velocidad_racha	12,506	0,154	elasticNetParam=1,0 regParam=1,0	NO
6	direccion_viento	12,024	0,161	elasticNetParam=1,0 regParam=1,0	NO
7	direccion_racha	12,52	0,156	elasticNetParam=1,0 regParam=1,0	NO
8	estacionAño	12,057	0,204	elasticNetParam=1,0 regParam=1,0	SI
9	tipoDia	11,861	0,229	elasticNetParam=1,0 regParam=1,0	SI
10	franjaHoraria	11,387	0,291	elasticNetParam=1,0 regParam=1,0	SI
11	estacionRecogida	10,920	0,344	elasticNetParam=1,0 regParam=1,0	SI

Tabla 7.6: Resultados de los experimentos de Regresión Lineal para la predicción de Dióxido de Nitrógeno

Modelo final: NO2 = temperatura + presión + estacionAño + tipoDia + franjaHoraria + estacionRecogida

NO2 - Árbol de decisión El algoritmo tarda alrededor de 14 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 8,094
- **R^2ajustado:** 0,609
- **Parámetro óptimo**: maxDepth=15

NO2 - Random Forest El algoritmo tarda alrededor de 13 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 10,225
- **R^2ajustado:** 0,376
- **Parámetro óptimo**: numTrees=5

CO - Regresión lineal El tiempo de procesado de los modelos está entre los 3 y 10 minutos.

	Variable probada	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	estacionRecogida	0,177	0,346	elasticNetParam=0,0 regParam=0,0	
2	temperatura	0,159	0,452	elasticNetParam=0,0 regParam=0,0	SI
3	presión	0,170	0,427	elasticNetParam=0,5 regParam=0,3	NO
4	humedad	0,161	0,470	elasticNetParam=0,5 regParam=0,3	SI
5	velocidad_viento	0,151	0,439	elasticNetParam=0,5 regParam=0,3	NO
6	velocidad_racha	0,152	0,447	elasticNetParam=0,5 regParam=0,3	NO
7	direccion_viento	0,150	0,463	elasticNetParam=0,5 regParam=0,3	NO
8	direccion_racha	0,154	0,441	elasticNetParam=0,5 regParam=0,3	NO
9	franjaHoraria	0,154	0,49	elasticNetParam=0,5 regParam=0,3	SI
10	tipoDia	0,154	0,486	elasticNetParam=0,5 regParam=0,3	NO
11	estacionAño	0,144	0,552	elasticNetParam=0,5 regParam=0,3	SI

Tabla 7.7: Resultados de los experimentos de Regresión Lineal para la predicción de Monóxido de Carbono

Modelo final: CO = estacionRecogida + temperatura + humedad + franjaHoraria + estacionAño.

CO - Árbol de decisión El algoritmo tarda alrededor de 5 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 0,121
- **R^2ajustado:** 0,671
- **Parámetro óptimo**: maxDepth=7

CO - Random Forest El algoritmo tarda alrededor de 4 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 0,13
- **R^2ajustado:** 0,621
- **Parámetro óptimo**: numTrees=5

PM10 - Regresión lineal En este y los siguientes modelos de predicción de PM10 se han eliminado los valores extremos, usando sólo aquellas filas donde PM10 $<$ 800 μg/m3. El tiempo de procesado de los modelos está entre los 5 y 10 minutos.

	Variable probada	RMSE (menos mejor)	R^2 adj (más mejor)	Parámetros óptimos	Mejora significativa
1	estacionRecogida	15,388	0,094	elasticNetParam=0,0 regParam=0,1	
2	velocidad_viento	14,553	0,112	elasticNetParam=1,0 regParam=1,0	SI
3	velocidad_racha	14,398	0,131	elasticNetParam=1,0 regParam=1,0	SI
4	precipitación	14,552	0,129	elasticNetParam=1,0 regParam=1,0	NO
5	presión	14,573	0,144	elasticNetParam=1,0 regParam=1,0	SI
6	direccion_viento	14,488	0,15	elasticNetParam=1,0 regParam=1,0	SI
7	direccion_racha	14,452	0,15	elasticNetParam=1,0 regParam=1,0	NO
8	temperatura	14,439	0,151	elasticNetParam=1,0 regParam=1,0	SI
9	humedad	14,37	0,159	elasticNetParam=1,0 regParam=1,0	SI
10	franjaHoraria	14,345	0,161	elasticNetParam=1,0 regParam=1,0	SI
11	tipoDia	14,297	0,166	elasticNetParam=1,0 regParam=1,0	SI
12	estacionAño	13,718	0,232	elasticNetParam=1,0 regParam=1,0	SI

Tabla 7.8: Resultados de los experimentos de Regresión Lineal para la predicción de partículas de diámetro inferior a 10μm

Modelo final: PM10 = estacionRecogida + velocidad_viento + velocidad_racha + presión + direccion_viento + temperatura + humedad + franjaHoraria + tipoDia + estacionAño

PM10 - Árbol de decisión El algoritmo tarda alrededor de 11 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 9,957
- **R^2ajustado:** 0,52
- **Parámetro óptimo**: maxDepth=5

PM10 - Random Forest El algoritmo tarda alrededor de 20 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 11,78
- **R^2ajustado:** 0,329
- **Parámetro óptimo**: numTrees=20

PM2,5 - Regresión lineal En este y los siguientes modelos de predicción de PM2,5 se han eliminado los valores extremos, usando sólo aquellas filas donde PM2,5 $<$ 800 µg/m3. El tiempo de procesado de los modelos está entre los 6 y 40 minutos.

	Variable probada	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	estacionRecogida	7,923	0,028	elasticNetParam=0,0 regParam=0,0	
2	presión	7,052	0,153	elasticNetParam=1,0 regParam=1,0	SI
3	velocidad_racha	6,669	0,199	elasticNetParam=1,0 regParam=1,0	SI
4	velocidad_viento	7,066	0,156	elasticNetParam=1,0 regParam=1,0	NO
5	temperatura	6,733	0,212	elasticNetParam=1,0 regParam=1,0	SI
6	precipitacion	6,734	0,211	elasticNetParam=1,0 regParam=1,0	NO
7	direccion_racha	6,732	0,212	elasticNetParam=1,0 regParam=1,0	NO
8	direccion_viento	7,124	0,179	elasticNetParam=1,0 regParam=1,0	NO
9	humedad	6,694	0,215	elasticNetParam=1,0 regParam=1,0	SI
10	tipoDia	6,673	0,219	elasticNetParam=1,0 regParam=1,0	SI
11	franjaHoraria	6,64	0,225	elasticNetParam=1,0 regParam=1,0	SI
12	estacionAño	6,549	0,245	elasticNetParam=1,0 regParam=1,0	SI

Tabla 7.9: Resultados de los experimentos de Regresión Lineal para la predicción de partículas de diámetro inferior a 2,5μm

Modelo final: PM2,5 = estacionRecogida + presión + velocidad_racha + temperatura + humedad + tipoDia + franjaHoraria + estacionAño

PM2,5 - Árbol de decisión El algoritmo tarda alrededor de 10 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 5,434
- **R^2ajustado:** 0,466
- **Parámetro óptimo**: maxDepth=27

PM2,5 - Random Forest El algoritmo tarda alrededor de 7 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 6,113
- **R^2ajustado:** 0,324
- **Parámetro óptimo**: numTrees=320

SO2- Regresión lineal En este y los siguientes modelos de predicción de SO2 se han eliminado los valores extremos, usando sólo aquellas filas donde SO2 $<$ 100 µg/m3. El tiempo de procesado de los modelos está entre los 3 y 20 minutos.

	Variable probada	RMSE (menos mejor)	R^2 adj (más mejor)	Parámetros óptimos	Mejora significativa
1	estacionRecogida	4,711	0,007	elasticNetParam=0,0 regParam=0,0	
2	temperatura	4,34	0,027	elasticNetParam=0,0 regParam=0,0	SI
3	humedad	5,281	0,013	elasticNetParam=0,7 regParam=1,0	NO
4	direccion_viento	4,275	0,035	elasticNetParam=1,0 regParam=1,0	SI
5	direccion_racha	4,86	0,037	elasticNetParam=1,0 regParam=1,0	SI
6	presión	4,727	0,039	elasticNetParam=0,7 regParam=1,0	SI
7	precipitacion	4,726	0,038	elasticNetParam=0,7 regParam=1,0	NO
8	velocidad_viento	4,694	0,052	elasticNetParam=0,7 regParam=1,0	SI
9	velocidad_racha	4,6	0,087	elasticNetParam=0,7 regParam=1,0	SI
10	tipoDia	4,6	0,085	elasticNetParam=0,7 regParam=1,0	NO
11	estacionAño	4,529	0,113	elasticNetParam=1,0 regParam=1,0	SI
12	franjaHoraria	4,512	0,116	elasticNetParam=1,0 regParam=1,0	SI

Tabla 7.10: Resultados de los experimentos de Regresión Lineal para la predicción de Dióxido de azufre

Modelo final: SO2 = idEstacion + temperatura + direccion_viento + direccion_racha + presión + velocidad_viento + velocidad_racha + estacionAño + franjaHoraria

SO2 - Árbol de decisión El algoritmo tarda alrededor de 5 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 3,067
- **R^2ajustado:** 0,584
- **Parámetro óptimo**: maxDepth=30

SO2 - Random Forest El algoritmo tarda alrededor de 9 minutos en finalizar el entrenamiento. Se obtiene el siguiente resultado:

- **RMSE:** 3,532
- **R^2ajustado:** 0,449
- **Parámetro óptimo**: numTrees=10

El mejor modelo para predecir todos los contaminantes es el árbol de decisión. Para cada contaminante el mejor modelo tiene las siguientes métricas:

Contaminante	R^2 ajustado	RMSE
O3	0,847	10,433
NO	0,742	7,533
CO	0,671	0,121
NO2	0,609	8,094
SO2	0,584	3,067
PM10	0,52	9,957
PM2,5	0,446	5,434

Tabla 7.11: Métricas resultantes de los modelos entrenados

7.2.2. Mejora del modelo utilizando predicciones

Hasta ahora los modelos han sido entrenados utilizando solo variables meteorológicas, de estación de recogida y relacionadas con el día y franja horaria de recogida. Sin embargo, sabemos que existen relaciones fuertes entre los propios contaminantes (por ejemplo, el NO2 se produce por interacción entre O3 y NO, por lo que es lógico pensar que estos dos contaminantes pueden ser valiosos para predecir NO2). Una primera aproximación podría ser utilizar datos reales de la hora anterior para predecir los datos de la hora actual, sin embargo, la fuente de datos de contaminantes que utilizamos no está disponible a tiempo real, sino que publica los datos de las últimas 24h, por lo que este método no sería posible.

Otra alternativa es utilizar los valores predichos de contaminantes. Estos datos no se corresponderán exactamente con el dato de contaminante real, sin embargo en algunos casos, como el Ozono, se asemejará lo suficiente. Un problema de esta aproximación es que no todas las estaciones de medida miden todos los contaminantes, por tanto por ejemplo al intentar predecir NO2 utilizando O3, sólo podremos predecirlo en aquellas estaciones donde se mida tanto NO2 como O3. Para remediar esto se ha utilizado la media en todo Valladolid para cada contaminante a la hora de introducirlo a los modelos.

Tras analizar las relaciones entre contaminantes y priorizar la utilización de salidas con un R^2ajustado alto, se propone añadir los contaminantes al modelo de la siguiente manera:

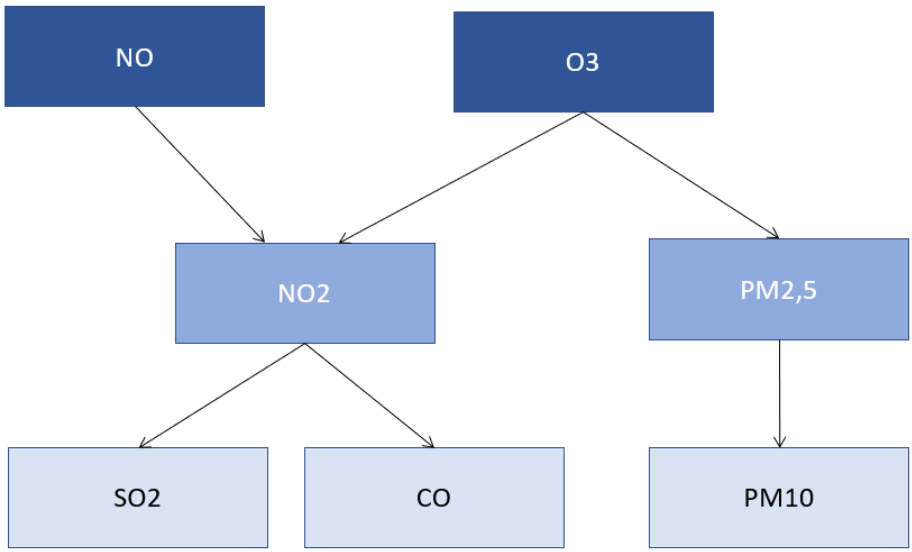


Figura 7.10: Mejora de los modelos utilizando predicciones

NO2					
	Modelo	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	temperatura + presión + estacionAño + tipoDia + franjaHoraria + estacionRecogida + NO_predicho	10,089	0,425	elasticNetParam = 0,0 regParam = 0,0	SI
2	temperatura + presion + estacionAño + tipoDia + franjaHoraria + estacionRecogida + NO_predicho + O3_predicho	9,228	0,519	elasticNetParam = 1,0 regParam = 1,0	
3	Árbol de decisión con todas las variables	7,605	0,673	maxDepth = 11	
4	Random Forest con todas las variables	8,351	0,606	numTrees = 5	

Tabla 7.12: Resultados de la segunda fase de experimentos para la predicción de Dióxido de Nitrógeno

PM2,5					
	Modelo	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	estacionRecogida + presión + velocidad_racha + temperatura + humedad + tipoDia + franjaHoraria + estacionAño + O3_predicho	6,172	0,285	elasticNetParam = 0,0 regParam = 0,0	NO
3	Árbol de decisión con todas las variables	5,124	0,444	maxDepth = 11	
4	Random Forest con todas las variables	5,455	0,365	numTrees = 320	

Tabla 7.13: Resultados de la segunda fase de experimentos para la predicción de partículas de diámetro inferior a 2,5μm

Al no haber una mejora significativa con estos nuevos modelos, el modelo utilizado para predecir el PM2,5 que servirá de entrada a los siguientes modelos es el que no incluye la predicción de O3.

SO2					
	Modelo	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	idEstacion + temperatura + direccion_viento + direccion_racha + presión + velocidad_viento + velocidad_racha + estacionAño + franjaHoraria + NO2_predicho	4,766	0,131	elasticNetParam = 0,0 regParam = 0,0	SI
3	Árbol de decisión con todas las variables	3,165	0,617	maxDepth = 27	
4	Random Forest con todas las variables	3,59	0,507	numTrees = 5	

Tabla 7.14: Resultados de la segunda fase de experimentos para la predicción de Dióxido de azufre

CO					
	Modelo	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	estacionRecogida + temperatura + humedad + franjaHoraria + estacionAño + NO2_predicho	0,148	0,524	elasticNetParam = 0,5 regParam = 0,3	SI
3	Árbol de decisión con todas las variables	0,109	0,718	maxDepth = 9	
4	Random Forest con todas las variables	0,121	0,652	numTrees = 160	

Tabla 7.15: Resultados de la segunda fase de experimentos para la predicción de Monóxido de carbono

PM10					
	Modelo	**RMSE (menos mejor)**	**R^2 adj (más mejor)**	**Parámetros óptimos**	**Mejora significativa**
1	estacionRecogida + velocidad_viento + velocidad_racha + presión + direccion_viento + temperatura + humedad + franjaHoraria + tipoDia + estacionAño + PM25_predicho	11,014	0,453	elasticNetParam = 0,0 regParam = 0,0	SI
3	Árbol de decisión con todas las variables	8,987	0,616	maxDepth = 30	
4	Random Forest con todas las variables	10,096	0,515	numTrees = 5	

Tabla 7.16: Resultados de la segunda fase de experimentos para la predicción de partículas de diámetro inferior a 10μm

Como se puede observar, todos los modelos menos el que predice PM2,5 experimentan una mejora significativa en su ajuste a los valores reales. Los mejores modelos siguen siendo aquellos que utilizan el árbol de decisión como algoritmo de regresión, y las mejores métricas logradas son:

Contaminante	R^2 ajustado	RMSE
O3	0,847	10,433
NO	0,742	7,533
CO	0,718	0,109
NO2	0,673	7,605
SO2	0,617	3,165
PM10	0,616	8,987
PM2,5	0,446	5,434

Tabla 7.17: Métricas resultantes de los modelos mejorados

Para estos últimos modelos, se trazan también los gráficos de predichos contra observados y residuos (observados - predichos) contra observados. Esto nos da una idea de si existen instancias con errores extremos en la predicción, o si hay una correlación entre los residuos y los valores reales observados.

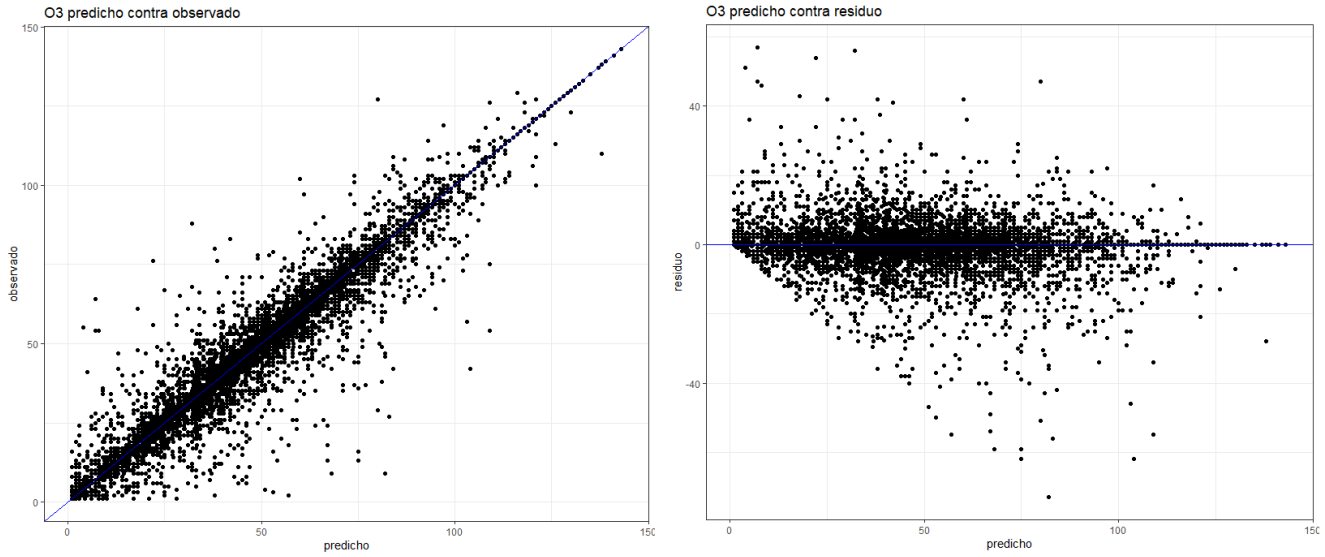


Figura 7.11: Gráficos de residuos, O3

Se puede observar un ajuste bueno de los valores predichos a los reales, así como una nube bastante aleatoria en los predichos contra residuos, sin relación aparente entre ambos valores.

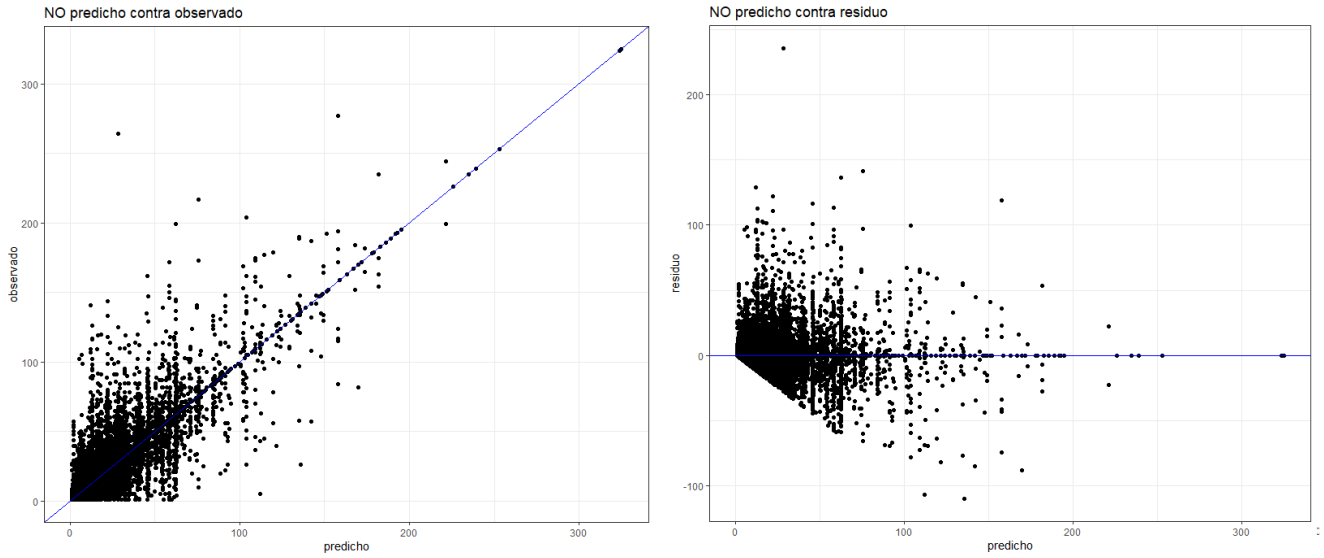


Figura 7.12: Gráficos de residuos, NO

El ajuste de los valores para el NO es visiblemente peor, además es claro un patrón en los residuos, con los residuos por debajo de 0 aumentando según aumenta el valor predicho. Los valores por encima de 0 parecen ser más aleatorios. Este resultado nos indica que nos encontramos con un modelo que se comporta peor cuando tiene que predecir valores medios que aquellos más bajos/altos.

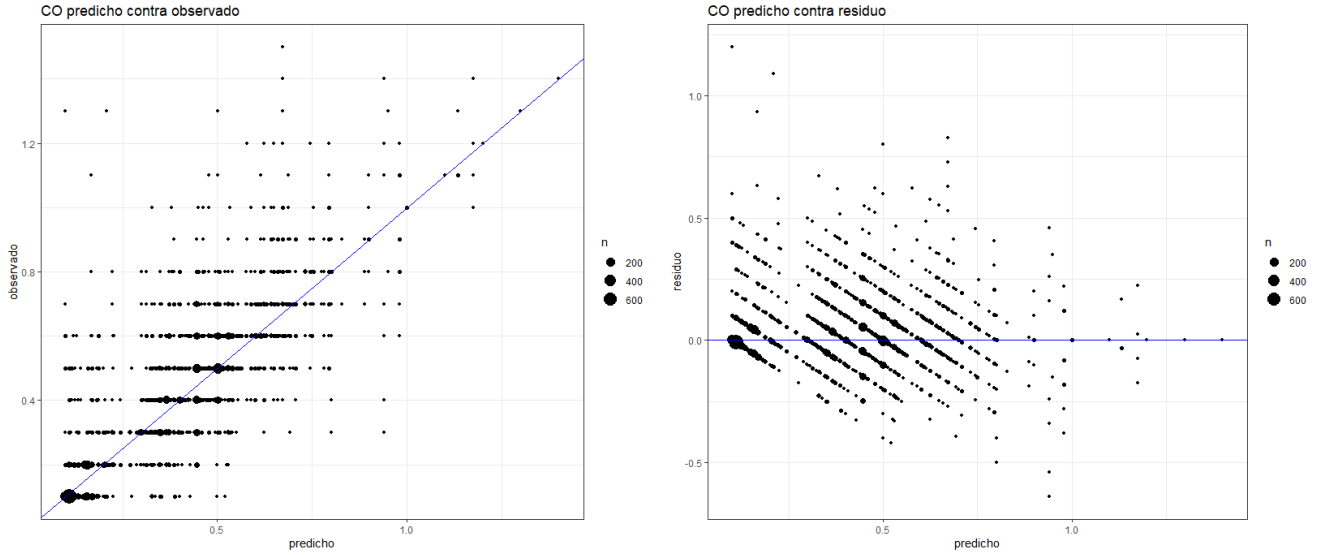


Figura 7.13: Gráficos de residuos, CO

Debido al relativamente pequeño rango de valores posibles de CO, los gráficos representan la cuenta del número

de puntos en cada coordenada mediante su área. Así, se puede ver que el ajuste de los valores es algo peor, aunque los valores más frecuentes siguen ocurriendo cerca de la recta y=x. Respecto a predichos contra residuos, la nube de puntos parece bastante aleatoria.

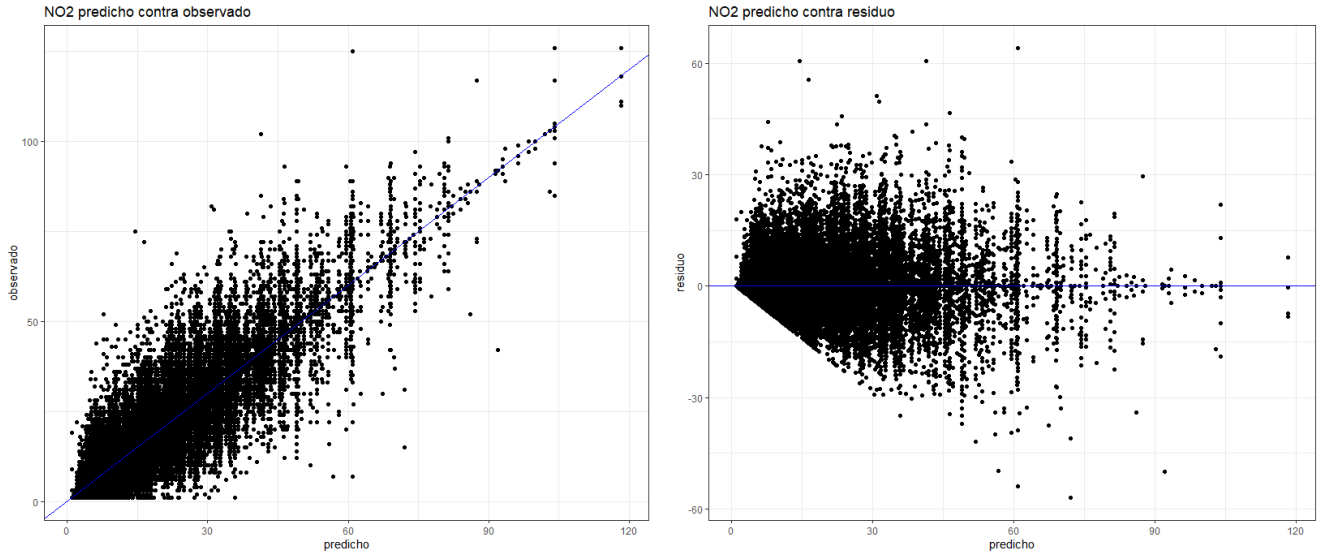


Figura 7.14: Gráficos de residuos, NO2

El NO2 presenta patrones muy similares al NO, aunque el ajuste del predicho al observado es peor.

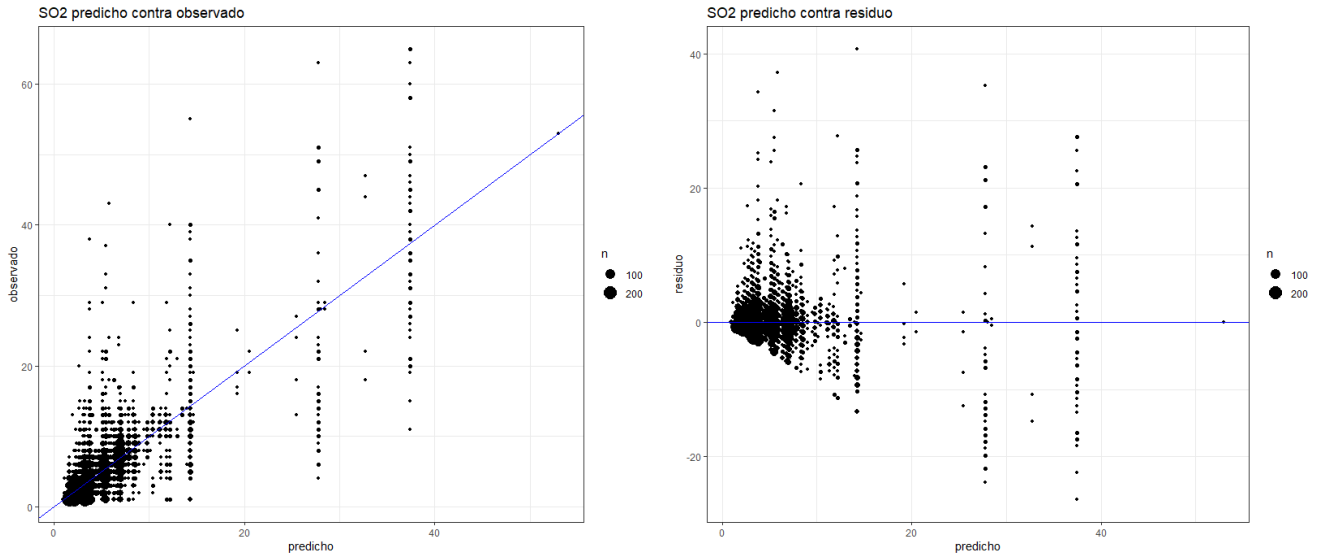


Figura 7.15: Gráficos de residuos, SO2

El modelo de SO2 parece predecir el mismo valor para un rango de valores observados muy amplio, aunque esto parece ocurrir solo cuando dicho valor predicho es alto. Esta inclinación hacia valores más incorrectos cuando el valor predicho es más alto se puede apreciar también en la nube de puntos de residuo contra predicho.

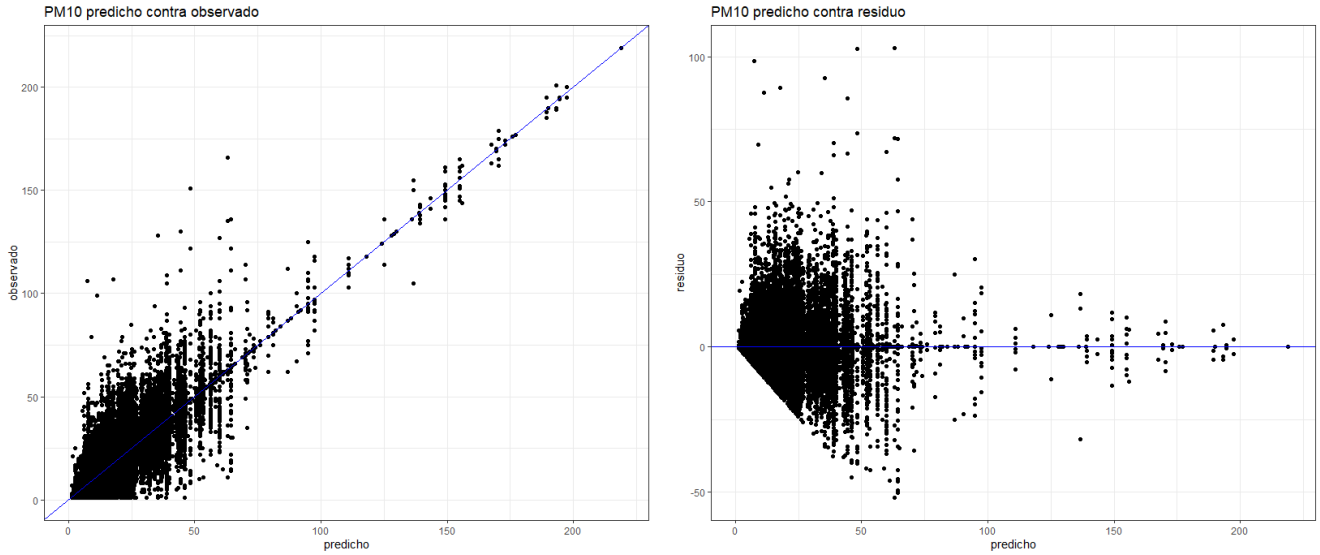


Figura 7.16: Gráficos de residuos, PM10

De nuevo ambos gráficos y análisis es asimilable a los resultantes para NO y NO2.

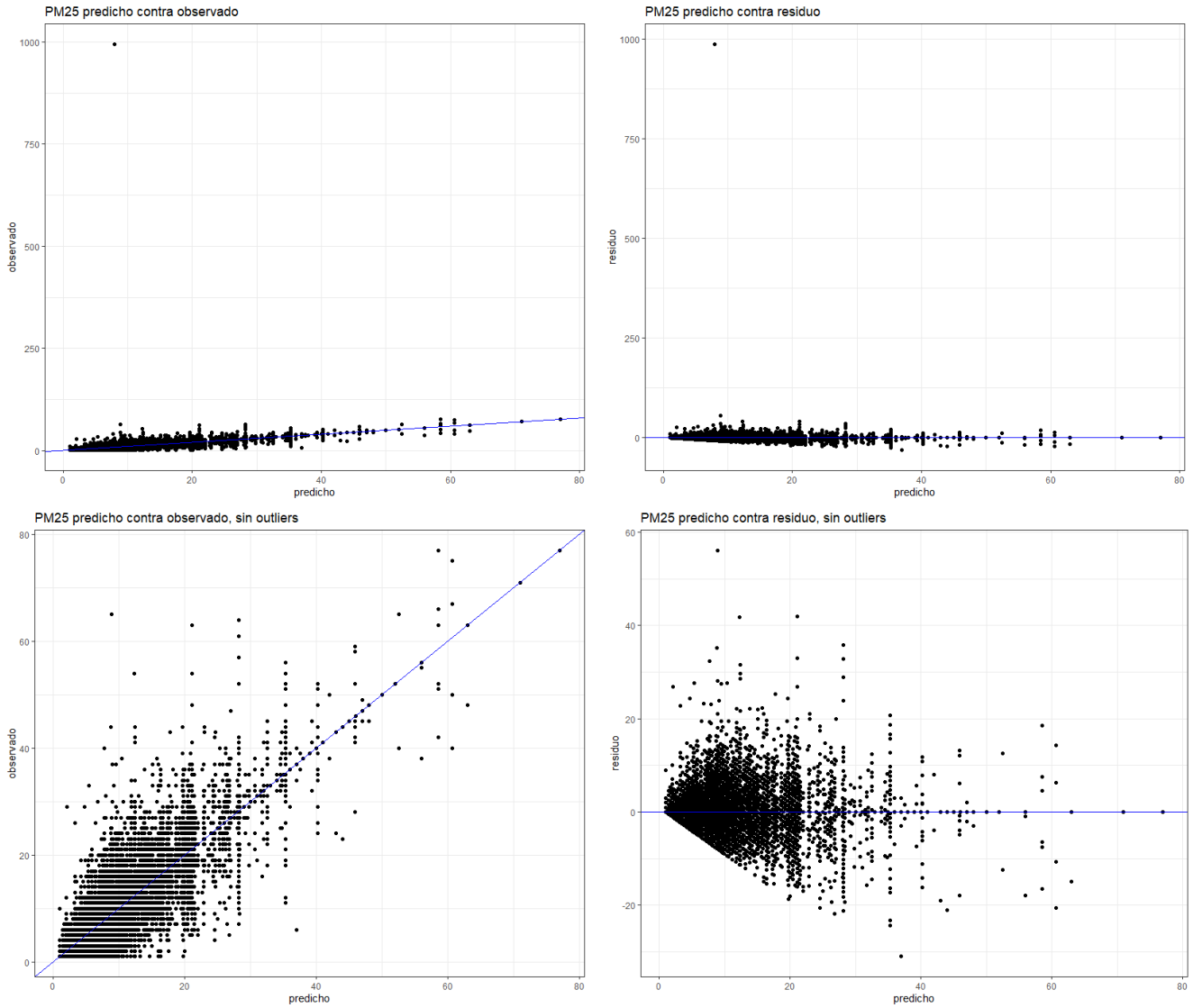


Figura 7.17: Gráficos de residuos, PM2,5

El modelo que predice PM25 se ve muy afectado por el valor extremo que sabemos que existe en los datos, ya que, como es de esperar, el modelo es incapaz de predecirlo correctamente. Eliminando dicho valor nos encontramos de nuevo con similitudes en la nube de puntos del gráfico de predicho contra residuos a NO, NO2 y PM10.

Capítulo 8

Conclusiones y trabajo futuro

La problemática del control de la calidad del aire está aún por resolver en la mayoría de regiones del mundo, incluida la ciudad de Valladolid, donde grandes concentraciones de Ozono continuaron estando por encima de los umbrales marcados por la Organización Mundial de la Salud durante el año 2019. Además no se dispone de ninguna herramienta a nivel local que ofrezca vigilancia y predicción de los contaminantes presentes en la provincia, aunque los datos disponibles son suficientes para un proyecto de este tipo.

En efecto, los datos de Valladolid se integran en múltiples aplicaciones a nivel internacional, aunque su carácter general impide dar usos más específicos a dichas herramientas. De igual manera existen múltiples modelos que tratan de predecir la calidad del aire en toda Europa o incluso en todo el mundo, y en este trabajo tratamos de utilizar algoritmos Big Data simples para realizar dicha tarea a nivel local.

El resultado del trabajo es un Data Lake complejo que integra múltiples fuentes de datos aprovechando la versatilidad de las numerosas herramientas disponibles tanto en el entorno Hadoop como otro software tradicional en ciencia de los datos, y cuyo carácter Big Data se completa con una visualización y predicción de los datos en un sistema de explotación semejante a lo que podría ser un servidor en producción real.

Las tareas exploratorias de los datos realizadas confirman de manera práctica algunos factores que se afirman en la teoría, como la presencia de patrones en los Óxidos de Carbono compatibles con los horarios de salida y entrada de trabajadores a las fábricas colindantes, o el aumento significativo del Ozono durante momentos de mayor insolación.

En efecto, los experimentos de predicción demuestran que, utilizando variables completamente públicas es posible predecir los valores futuros de la mayoría de contaminantes con un nivel de precisión alto. En efecto, como podemos observar en la Figura 2.9, BreezoMeter consigue valores de RMSE comparables, y en algunos casos peores, que los obtenidos con nuestros modelos (Tabla 7.17). Estos resultados nos muestran que los modelos producidos se asemejan a ofertas comerciales existentes, lo que refuerza la importancia que tienen las iniciativas Open Data en gobiernos e instituciones locales.

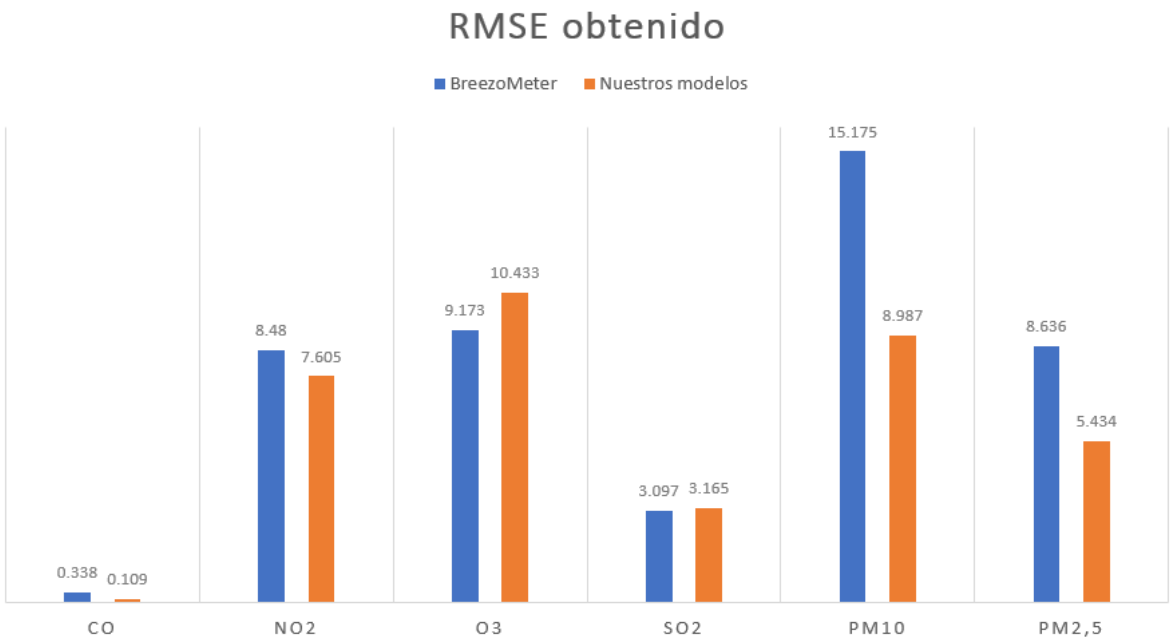


Figura 8.1: Comparativa entre los resultados de BreezoMeter y los de nuestros modelos

8.1. Trabajo Futuro

Un proyecto Big Data se asocia, naturalmente, con grandes volúmenes de datos. En nuestro caso, ya que únicamente utilizamos datos de las estaciones en la provincia de Valladolid, el dataset más extenso contará con menos de un millón de mediciones. Sin embargo, en este trabajo se ha demostrado la utilidad del Data Lake en un proyecto de este tipo, y sería posible integrar datos de todo Castilla y León con muy pocos cambios, lo que multiplicaría de manera significativa la cantidad de datos.

Respecto a la predicción de datos, se han probado solamente tres algoritmos, aunque existen muchas otras opciones para la predicción de este tipo de datos. Una opción interesante aunque sin implementación nativamente en Spark ML son las Redes Neuronales Recurrentes, muy utilizadas para la predicción de series temporales. Las Redes Neuronales Recurrentes han sido utilizadas con éxito para la predicción de PM10 [35], aunque son computacionalmente muy costosas, por lo que es difícil implementarlas para Big Data de manera escalable. Sin embargo con un subconjunto de datos reducido sería posible comprobar la eficacia de dichos algoritmos.

También se ha considerado utilizar la implementación en Spark del Gradient Boosted Tree, un algoritmo que entrena múltiples árboles de manera similar al Random Forest pero utilizando otra técnica conocida como *boosting*. Se configuró un experimento con dicho algoritmo, sin embargo el tiempo de entrenamiento superaba las varias decenas de horas por modelo, lo que lo hacía inviable para este trabajo, sin embargo, de manera similar a las redes neuronales, podría ser posible obtener un modelo viable utilizando un subconjunto de los datos existentes.

Bibliografía

[1] Subdirección General de Aire Limpio y Sostenibilidad Industrial. Evaluación de la calidad del aire en españa. Technical report, Ministerio para la Transición Ecológica y el Reto Demográfico, 2019. Disponible en https://www.miteco.gob.es/images/es/informeevaluacioncalidadaireespana2019_tcm30-510616.pdf. Visitado por última vez el 15 de Septiembre de 2020.

[2] Scientia. One, two, three, breathe. Technical report, 2018. Disponible en https://info.breezometer.com/hubfs/Press%20Kit/Scientia%20-%20Technology%20Paper%20on%20BreezoMeter.pdf. Visitado por última vez el 13 de Septiembre de 2020.

[3] Organización mundial de la salud. Ambient (outdoor) air pollution, Mayo 2018. Disponible en https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health. Visitado por última vez el 15 de Septiembre de 2020.

[4] Organización mundial de la salud. Health and the environment: addressing the health impact of air pollution. Mayo 2015. Disponible en https://apps.who.int/iris/bitstream/handle/10665/253237/A68_R8-en.pdf?sequence=1&isAllowed=y. Visitado por última vez el 15 de Septiembre de 2020.

[5] Naciones Unidas. Transformar nuestro mundo: la agenda 2030 para el desarrollo sostenible. Disponible en https://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1&Lang=S. Visitado por última vez el 18 de Septiembre de 2020.

[6] Organización mundial de la salud. Objetivo 11: Lograr que las ciudades sean mÃ¡s inclusivas, seguras, resilientes y sostenibles. Disponible en https://www.un.org/sustainabledevelopment/es/cities/. Visitado por última vez el 15 de Septiembre de 2020.

[7] D. V. Valladolid supera los límites de contaminación en 2019. *El día de Valladolid*, Enero 2020. Disponible en https://www.eldiadevalladolid.com/noticia/Z2FF2DBF1-CAFD-6483-7C219B359A608EE6/202001/Valladolid-supera-los-limites-de-contaminacion-en-2019. Visitado por última vez el 15 de Septiembre de 2020.

[8] J. S. Carlos disdier: "si no se tomaran medidas contra la polución se acabarían saturando nuestros hospitales". *El Norte de Castilla*, Febrero 2019. Disponible en https://www.elnortedecastilla.es/valladolid/carlos-didier-tomaran-20190228213445-nt.html. Visitado por última vez el 15 de Septiembre de 2020.

[9] D.H.F. Liu and B.G. Lipták. *Air Pollution*. Lewis Publishers, 2000.

[10] Ministerio Para la Transición Ecológica y el Reto Demográfico Gobierno de España. Problemática ambiental y contaminantes. Disponible en `https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/emisiones/prob-amb/default.aspx`. Visitado por última vez el 13 de Septiembre de 2020.

[11] Ministerio Para la Transición Ecológica y el Reto Demográfico Gobierno de España. Monóxido de carbono. Disponible en `https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/salud/monoxido-carbono.aspx`. Visitado por última vez el 13 de Septiembre de 2020.

[12] Bert Brunekreef and Stephen T Holgate. Air pollution and health. *The lancet*, 360(9341):1233–1242, 2002.

[13] Jairo Téllez, Alba Ii, and Álvaro Fajardo. Contaminación por monóxido de carbono: un problema de salud ambiental. *Revista de Salud Pública*, 8:108–117, 04 2006.

[14] Victoria Bermejo, Rocío Alonso, Susana Elvira, Isaura Rábago, and Marta García Vivanco. *El ozono troposférico y sus efectos en la vegetación*. 01 2009.

[15] Ayuntamiento de Valladolid. Â¿qué es la rccava? Disponible en `https://www.valladolid.es/es/rccava/rccava`. Visitado por última vez el 13 de Septiembre de 2020.

[16] Comisión Europea. Air quality in europe: Ensemble modelling. Disponible en `https://www.regional.atmosphere.copernicus.eu/`. Visitado por última vez el 13 de Septiembre de 2020.

[17] Comisión Europea. Atmosphere monitoring service, Septiembre 2016. Disponible en `https://www.copernicus.eu/sites/default/files/documents/Copernicus_AtmosphereMonitoring_Feb2017.pdf`. Visitado por última vez el 13 de Septiembre de 2020.

[18] Victoria J. Hodge, Simon O'Keefe, and Jim Austin. Hadoop neural network for parallel and distributed feature selection. *Neural Networks*, 78:24 – 35, 2016.

[19] Yang Liu, Youbo Liu, Junyong Liu, Maozhen Li, Tingjian Liu, Gareth Taylor, and Kunyu Zuo. A mapreduce based high performance neural network in enabling fast stability assessment of power systems. *Mathematical Problems in Engineering*, 2017, 2017.

[20] Pratap Dangeti. *Statistics for machine learning*. Packt Publishing Ltd, 2017.

[21] Danny Varghese. Comparative study on classic machine learning algorithms. Disponible en `https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222`. Visitado por última vez el 13 de Septiembre de 2020.

[22] A. Gorelik. *The Enterprise Big Data Lake: Delivering the Promise of Big Data and Data Science*. O'Reilly Media, 2019.

[23] Matillion Ltd. Essential guide to data lakes - designing data lakes to optimize analytics. Technical report, 2019. Disponible en http://pages.matillion.com/rs/992-UIW-731/images/2019C2%20-%20Data%20Lakes%20eBook.pdf. Visitado por última vez el 13 de Septiembre de 2020.

[24] The Apache Software Foundation. Hdfs architecture guide. Disponible en https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html. Visitado por última vez el 13 de Septiembre de 2020.

[25] The Apache Software Foundation. Apache hive. Disponible en https://cwiki.apache.org/confluence/display/Hive/Home. Visitado por última vez el 13 de Septiembre de 2020.

[26] The Apache Software Foundation. Spark overview. Disponible en https://spark.apache.org/docs/latest/. Visitado por última vez el 13 de Septiembre de 2020.

[27] The Apache Software Foundation. Apache sqoop. Disponible en https://sqoop.apache.org/. Visitado por última vez el 13 de Septiembre de 2020.

[28] The Apache Software Foundation. Apache oozie workflow scheduler for hadoop. Disponible en https://oozie.apache.org/. Visitado por última vez el 13 de Septiembre de 2020.

[29] The Apache Software Foundation. Spark ml programming guide. Disponible en https://spark.apache.org/docs/1.2.2/ml-guide.html. Visitado por última vez el 13 de Septiembre de 2020.

[30] 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 (53 págs.) Disponible en https://www.boe.es/buscar/doc.php?id=BOE-A-2011-1645. Visitado por última vez el 13 de Septiembre de 2020.

[31] Ministerio de la Presidencia y para las Administraciones Territoriales. Real decreto 39/2017, de 27 de enero, por el que se modifica el real decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire, Enero 2017. Publicado en BOE núm. 24, de 28 de enero de 2017, páginas 6918 a 6930 (13 págs.) Disponible en https://www.boe.es/buscar/doc.php?id=BOE-A-2017-914. Visitado por última vez el 13 de Septiembre de 2020.

[32] Organización mundial de la salud y otros. Guías de calidad del aire de la oms relativas al material particulado, el ozono, el dióxido de nitrógeno y el dióxido de azufre: actualización mundial 2005. Technical report, Ginebra: Organización Mundial de la Salud, 2006.

[33] The Apache Software Foundation. Oozie coordinator specification. Disponible en https://oozie.apache.org/docs/3.1.3-incubating/CoordinatorFunctionalSpec.html. Visitado por última vez el 13 de Septiembre de 2020.

[34] A. Kirk. *Data Visualization: A Successful Design Process*. Community experience distilled. Packt Pub., 2012.

[35] Athira V, Geetha P, Vinayakumar R, and Soman K P. Deepairnet: Applying recurrent networks for air quality prediction. *Procedia Computer Science*, 132:1394 – 1403, 2018. International Conference on Computational Intelligence and Data Science.

Apéndice A

Código de creación de tablas en Hive

```
CREATE DATABASE IF NOT EXISTS tfm;
USE tfm;

CREATE EXTERNAL TABLE IF NOT EXISTS festivos (
    fecha STRING )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/raw/Festivos' TBLPROPERTIES("skip.header.
    line.count"="1","serialization.encoding"='ISO-8859-1');

CREATE EXTERNAL TABLE IF NOT EXISTS estaciones (
    estacion STRING, localizacion STRING,
    provincia STRING, longitud STRING,
    latitud STRING, altitud STRING, operativa STRING )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/raw/Estaciones/detalles' TBLPROPERTIES("
    skip.header.line.count"="1","serialization.encoding"='ISO-8859-1');

CREATE EXTERNAL TABLE IF NOT EXISTS estaciones_ids (
    estacion STRING, id INT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/raw/Estaciones/ids' TBLPROPERTIES("skip.
    header.line.count"="1","serialization.encoding"='ISO-8859-1');

CREATE EXTERNAL TABLE IF NOT EXISTS calaire (
    no FLOAT, o3 FLOAT, nombreProvincia STRING, pm10 FLOAT, idEstacion INT,
    co FLOAT, no2 FLOAT, pst FLOAT, fecha STRING, pm25 FLOAT, codProvincia INT, so2
```

```
        FLOAT,
    nombreEstacion STRING, sh2 FLOAT, timestamp_value STRING )
PARTITIONED BY ( yymm STRING )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/raw/ESCO' TBLPROPERTIES("skip.header.line.
    count"="1","serialization.encoding"='ISO-8859-1');

CREATE EXTERNAL TABLE IF NOT EXISTS umbrales (
    contaminante STRING, temporalidad STRING, valor FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/raw/Umbrales' TBLPROPERTIES("skip.header.
    line.count"="1","serialization.encoding"='ISO-8859-1');

CREATE EXTERNAL TABLE IF NOT EXISTS meteo (
    vv FLOAT, tss20cm FLOAT, dmaxu FLOAT, pres FLOAT, tamax FLOAT, idema INT,
    lon FLOAT, hr FLOAT, dv FLOAT, prec FLOAT, tamin FLOAT, vmax FLOAT,lat FLOAT,
    vis FLOAT, dvu FLOAT, fint STRING, alt FLOAT, pres_nmar FLOAT, stdvv FLOAT,
    ta FLOAT, stddvu FLOAT, dmax FLOAT, inso FLOAT, tss5cm FLOAT, vvu FLOAT,
    vmaxu FLOAT, tpr FLOAT, pacutp FLOAT, ubi STRING, stdvvu FLOAT, ts FLOAT, stddv
        FLOAT )
PARTITIONED BY ( yymm STRING )  ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES
    TERMINATED BY '\n'  STORED AS TEXTFILE LOCATION '/user/esther/TFM/raw/AEMET'
    TBLPROPERTIES("skip.header.line.count"="1","serialization.encoding"='ISO-8859-1
    ');

CREATE EXTERNAL TABLE IF NOT EXISTS L0_festivos (
    fecha STRING  )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L0/festivos';

CREATE EXTERNAL TABLE IF NOT EXISTS L0_estaciones (
    nombre STRING, localizacion STRING, provincia STRING,
     longitud FLOAT, latitud FLOAT, altitud FLOAT, operativa BOOLEAN )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L0/estaciones';

CREATE EXTERNAL TABLE  IF NOT EXISTS L0_calaire (
    idEstacion INT, fecha STRING, hora INT,
    co FLOAT, no FLOAT, no2 FLOAT, o3 FLOAT,
     pm10 FLOAT, pm25 FLOAT, so2 FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
```

```
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L0/calaire';

CREATE EXTERNAL TABLE  IF NOT EXISTS L0_meteo (
    idEstacion INT, fecha STRING, hora INT,
    temperatura FLOAT, velocidad_viento FLOAT,
     direccion_viento INT, velocidad_racha FLOAT,
     direccion_racha INT, precipitacion FLOAT, presion FLOAT,
     humedad FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L0/meteo';

CREATE EXTERNAL TABLE IF NOT EXISTS L0_umbral (
    id STRING, contaminante STRING, temporalidad STRING, limite FLOAT ) ROW FORMAT
       DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
       TEXTFILE LOCATION '/user/esther/TFM/transformation/L0/umbral';

CREATE EXTERNAL TABLE IF NOT EXISTS L1_estacion_cal (
    id INT, nombre STRING, localizacion STRING, provincia STRING,
    longitud FLOAT, latitud FLOAT, altitud FLOAT, operativa BOOLEAN )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L1/estacion_cal';

CREATE EXTERNAL TABLE IF NOT EXISTS L1_estacion_meteo (
    id INT, nombre STRING, longitud FLOAT, latitud FLOAT, altitud FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L1/estacion_meteo';

CREATE EXTERNAL TABLE IF NOT EXISTS L2_superaumbral (
    idUmbral STRING, idEstacion INT, fecha STRING, hora INT, valor FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L2/superaumbral';

CREATE EXTERNAL TABLE IF NOT EXISTS L2_influencia (
    fecha_meteo STRING, hora_meteo INT, idEstacion_meteo INT,
    fecha_calaire STRING, hora_calaire INT, idEstacion_calaire INT,
    festivo BOOLEAN )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L2/influencia';

CREATE EXTERNAL TABLE IF NOT EXISTS temp_SuperaUmbral(
    idUmbral STRING, idEstacion INT, fecha STRING,
```

```
        hora STRING, valor FLOAT )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L3/temp_superaumbral';

CREATE EXTERNAL TABLE IF NOT EXISTS temp_MedicionCalaire(
    idEstacion STRING, fecha STRING, hora INT,
        CO FLOAT, NO FLOAT, NO2 FLOAT, O3 FLOAT,
    PM10 FLOAT, PM25 FLOAT, SO2 FLOAT, temperatura FLOAT,
        velocidad_viento FLOAT, direccion_viento INT,
    velocidad_racha FLOAT, direccion_racha INT,
    precipitacion FLOAT, presion FLOAT, humedad FLOAT,
    festivo BOOLEAN )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'  STORED AS
    TEXTFILE LOCATION '/user/esther/TFM/transformation/L3/temp_medicioncalaire';

CREATE TABLE IF NOT EXISTS DH_EstacionMeteo(
        id INT, nombre STRING, longitud FLOAT,
        latitud FLOAT, altitud FLOAT )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_MedicionMeteo(
        idEstacion STRING, fecha STRING, hora INT,
        temperatura FLOAT, velocidad_viento FLOAT,
        direccion_viento INT, velocidad_racha FLOAT,
        direccion_racha INT, precipitacion FLOAT,
        presion FLOAT, humedad FLOAT )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_EstacionCalAire(
        id INT, nombre STRING, localizacion STRING,
        provincia STRING, longitud FLOAT, latitud FLOAT,
        altitud FLOAT, operativa FLOAT )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_MedicionCalAire(
        idEstacion STRING, fecha STRING, hora INT,
        CO FLOAT, NO FLOAT, NO2 FLOAT, O3 FLOAT,
    PM10 FLOAT, PM25 FLOAT, SO2 FLOAT,
```

```
        temperatura FLOAT, velocidad_viento FLOAT,
        direccion_viento INT, velocidad_racha FLOAT,
        direccion_racha INT, precipitacion FLOAT,
        presion FLOAT, humedad FLOAT, festivo BOOLEAN )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_Influencia(
        fecha_meteo STRING, hora_meteo INT,
        idEstacion_meteo INT, fecha_calaire STRING,
        hora_calaire INT, idEstacion_calaire INT,
        festivo BOOLEAN )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_SuperaUmbral(
        idUmbral STRING, idEstacion INT, fecha STRING,
        hora STRING, valor FLOAT )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;

CREATE TABLE IF NOT EXISTS DH_Umbral(
        id STRING, contaminante STRING,
        temporalidad STRING,
        limite FLOAT )
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS
    TEXTFILE;
```

Apéndice B

Código de creación de tablas en la base de datos de explotación

```
CREATE TABLE EstacionCalAire(
 id int PRIMARY KEY NOT NULL,
  nombre varchar(25),
  localizacion varchar(25),
  provincia varchar(25) NOT NULL,
  longitud float,
  latitud float,
  altitud float,
  operativa bit NOT NULL
) GO

CREATE TABLE MedicionCalAire (
  idEstacion int NOT NULL,
  fecha CHAR(10) NOT NULL,
  hora int NOT NULL,
  CO float, NO FLOAT,
  NO2 FLOAT, O3 FLOAT,
  PM10 FLOAT, PM25 FLOAT,
  SO2 FLOAT, temperatura FLOAT,
  velocidad_viento FLOAT,
  direccion_viento INT,
  velocidad_racha FLOAT,
  direccion_racha INT,
  precipitacion FLOAT,
```

```
  presion FLOAT,  humedad FLOAT,
  festivo BIT,
  CONSTRAINT PK_medicioncalaire PRIMARY KEY (fecha, hora, idEstacion),
  CONSTRAINT FK_idEstacionCalAire FOREIGN KEY (idEstacion) REFERENCES
      EstacionCalAire(id)
)   GO

CREATE TABLE Umbral (
  id char(36) PRIMARY KEY NOT NULL,
  contaminante varchar(4) NOT NULL,
  temporalidad varchar(7) NOT NULL,
  limite FLOAT NOT NULL
) GO

CREATE TABLE SuperacionUmbral (
  idUmbral char(36) NOT NULL,
  idEstacion int NOT NULL,
  fecha CHAR(10) NOT NULL,
  hora int NOT NULL,
  valor FLOAT
  CONSTRAINT PK_superacionumbral PRIMARY KEY (idUmbral, fecha, hora, idEstacion),
  CONSTRAINT FK_umbral FOREIGN KEY (idUmbral) REFERENCES Umbral(id),
  CONSTRAINT FK_medicion FOREIGN KEY(fecha, hora, idEstacion) REFERENCES
      MedicionCalAire(fecha,hora,idEstacion)
)   GO
```