

Instituto Tecnológico Telefónica

Programa superior en Big Data



Efectos del confinamiento en la calidad del aire de la Comunidad de Madrid

Grupo 7

AUTORES:
Alfonso Gallardo
Raúl Hervás
Carmen Reina
Walter Ronceros
Susana Vara

Agradecimientos

En esta ocasión, queremos expresar nuestra más profunda condolencia a todas aquellas personas que se hayan visto afectadas de forma directa o indirecta por la pandemia producida por la enfermedad COVID-19.

"Decir que el hombre es una mezcla de fuerza y de debilidad, de luz y de ceguera, no es hacer su proceso: es definirlo."
(Denis Diderot)

Índice

RESUMEN:	4
1.- Introducción	5
2.- Objetivos y alcance de proyecto	6
2.1. Motivación del proyecto	6
2.2. Objetivos finales	6
2.3. Limitaciones	6
3.- Desarrollo	7
3.1 Metodología	7
3.2 Roles	9
3.3. Tecnologías empleadas	12
3.3.1 Python	12
3.3.2 Anaconda Navigator	12
3.3.3 Docker	12
3.3.4 Jupyter Notebook	13
3.3.5 Sypder	13
3.3.6 Elasticsearch	13
3.3.7 Databricks	13
3.4 Entorno Colaborativo	14
3.4.1 Google Colab	14
3.4.2 GitHub	15
3.4.2 Trello	16
4.- Fases del proyecto	17
4.1 Comprensión de Negocio	17
4.1.1 Sustancias contaminantes	18
4.1.2 Estaciones de monitorización de la calidad del aire del Ayuntamiento de Madrid y de la Comunidad de Madrid	18
4.1.3 Registro de Datos, Interprete de Ficheros de datos horarios, diarios y tiempo real	20
4.1.4 Índice de calidad del aire	21
4.1.5 Adaptación a la Directiva 2008/50/CE en España	25
4.1.6 Protocolo de Contaminación en Madrid	26
4.1.7 Cronología del COVID en España	28
4.1.8 Fases del Desconfinamiento	31
4.2 Comprensión de los datos: Data Engineering, poniendo orden en el caos de los datos	33
4.2.1 Fuentes de datos	33
4.3. Preparación de los datos: Análisis y unificación de los datasets	37
4.3.1. Carga de fuentes de datos	38
4.3.2. Comprensión y análisis descriptivo de los datos	41
4.3.3. Pre-procesado de datos	42
4.3.4. Análisis Descriptivo de los Datos	50
4.4 Modelado	67
4.4.1. Preparación del modelo	67

4.4.2. Desarrollo de los modelos	70
4.4.3. Mejora de precisión del modelo	75
4.5 Evaluación	76
4.6 Implementación	77
5.- Conclusión.....	81
6.- Anexos.....	84
6.1 ANEXO I. CÓDIGOS DE ESTACIONES	84
6.2 ANEXO II. MAGNITUDES, UNIDADES Y TÉCNICAS DE MEDIDA	85
6.3 ANEXO III. VALORES LIMITE.....	86
6.4 ANEXO IV. CÓDIGO PYTHON API AEMET.....	87
6.5 INFORMACION RELEVANTE DEL TRABAJO GRUPAL	89
Integrantes del Grupo 7.....	89
Limitaciones	89
Tecnologías 	89
Instrucciones para la reproducción del trabajo	90
Instrucciones para entorno de visualización.....	90
Índice de links externos	91
7.- Tabla de ilustraciones	92
8.- Bibliografía	95
9. Glosario	97

RESUMEN:

La salud pública, así como el cuidado del medioambiente son dos de los objetivos principales de los Gobiernos. Ambos pueden verse fuertemente impactados por altas tasas de contaminantes en el ambiente. Por ello, los Gobiernos han adecuado medidas necesarias para su control y mejora, haciendo foco en la reducción de la contaminación en las ciudades.

Uno de los aspectos fundamentales para ello es la observación de las magnitudes interviniéntes en la calidad del aire. Debido a los cambios recientes de los usos de los vehículos, este trabajo está enfocado a analizar si ha habido un impacto positivo en la reducción de los niveles de la contaminación de la ciudad y comunidad de Madrid durante el confinamiento de la población debido al COVID-19, comparándolo con el mismo periodo de los años 2018 y 2019. Es por ello por lo que en este trabajo se evalúan todas las variables que intervienen en el índice de contaminación ambiental antes, durante y después del confinamiento. Otros de los aspectos relevantes que engloba este trabajo es la predicción de dicha contaminación a través de diferentes modelos predictivos y su comparación para identificar cual es el que mejor predice la contaminación atmosférica en Madrid.

1.- Introducción

La contaminación del aire impacta significativamente en la salud de la población europea, especialmente en las zonas urbanas. Asimismo, tiene un considerable impacto económico, puesto que aumenta los costes sanitarios derivados de las enfermedades que provoca. Esto ha llevado a la calidad del aire a convertirse en una de las principales cuestiones políticas de las últimas décadas en el ámbito europeo.

La importancia de este tema se ve reflejada en las múltiples Directivas Europeas que tratan de establecer unos mínimos de calidad de aire para proteger a la población de excesivos niveles de contaminación.

La progresiva regulación que se ha ido aprobando desde finales de los 70 se ha consolidado en la Directiva 2008/50/CE relativa a la calidad del aire ambiente y a una atmósfera más limpia en Europa. Haciendo una muy breve puntualización jurídica, una Directiva Europea es una norma de ámbito supranacional en la que se fija una serie de objetivos a alcanzar, pero deja libertad a los Estados miembros para elegir los medios que consideren más convenientes para alcanzarlos. Dicha norma comunitaria, en su artículo primero señala cuáles son sus objetivos:

- Definir y establecer objetivos de calidad del aire ambiente para evitar, prevenir o reducir los efectos nocivos para la salud humana y el medio ambiente en su conjunto.
- Evaluar la calidad del aire ambiente en los Estados miembros basándose en métodos y criterios comunes.
- Obtener información sobre la calidad del aire ambiente con el fin de ayudar a combatir la contaminación atmosférica y otros perjuicios y controlar la evolución a largo plazo y las mejoras resultantes de las medidas nacionales y comunitarias.
- Asegurar que esa información sobre calidad del aire ambiente se halla a disposición de los ciudadanos.
- Mantener la calidad del aire, cuando sea buena, y mejorarla en los demás casos.
- Fomentar el incremento de la cooperación entre los Estados miembros para reducir la contaminación atmosférica.

Los contaminantes más serios en términos de deterioro para la salud son las partículas en suspensión PM2.5 y los óxidos de Nitrógeno NOx.

Durante la paralización del país debido al COVID'19 en el mes de marzo es evidente que, sin actividad humana, la naturaleza se abre camino. Durante estas largas semanas de confinamiento se han visto noticias donde animales nunca antes vistos en ciudades se aventuraban a hacer turismo, el olor en la

ciudad cambiaba radicalmente pudiéndose percibir olor a vegetación incluso la falta de ruido desorientaba a ciertos animales de ciudad acostumbrados al estrés y contaminación acústica diaria.

Con este estudio queremos ver qué puntos de inflexión pueden verse en la calidad del aire de la Comunidad de Madrid (de ahora en adelante CAM) en las fechas de confinamiento para tratar de obtener el tiempo que tarda la naturaleza en llegar a la máxima calidad de aire y el ser humano en devolverla a sus niveles "habituales".

2.- Objetivos y alcance de proyecto

2.1. Motivación del proyecto

Gracias a toda la información recogida de agentes contaminantes por las estaciones situadas por toda la CAM y estando al alcance de todos, así como información meteorológica como velocidad y dirección del viento, precipitaciones o radiación solar entre otros queremos contrastarla para tratar de encontrar alguna relación directa entre ellas que pueda ayudar a predecir los niveles de calidad del aire a futuro.

2.2. Objetivos finales

El objeto de este trabajo es la comprensión de las magnitudes y variables que arrojan datos con relación a la calidad del aire y contaminación atmosférica de la comunidad de Madrid, incluyendo la ciudad de Madrid, durante el periodo de confinamiento vivido en España a raíz del Decreto que sometió a confinamiento a todo el país, visualización los cambios en los datos recogidos por la red de estaciones que el Ayuntamiento tiene desplegadas en la capital.

Para ello, realizaremos un modelo predictivo, que obteniendo los datos de las APIs donde se publica la información de la CAM (incluido el Ayuntamiento de Madrid), desde enero 2020 hasta final del estado de alarma, sea capaz de predecir o estimar la calidad del aire una vez vuelta la actividad normal.

2.3. Limitaciones

Para este estudio damos por hecho ciertas limitaciones insalvables como son:

- No todas las estaciones contienen medición de todos los agentes contaminantes.

- Las mediciones meteorológicas no se pueden prever.
- El índice ICA (Índice Calidad del Aire) se obtiene de la medición más adversa de 5 agentes contaminantes por lo que no puede utilizarse para buscar correlaciones directas.
- Las magnitudes meteorológicas están sometidas a numerosos cambios adversos, que pueden derivar por motivos multifactoriales y pueden no atender a una sola causa, en este proyecto abarcamos principalmente los cambios referentes al confinamiento debidos a la pandemia del COVID-19.

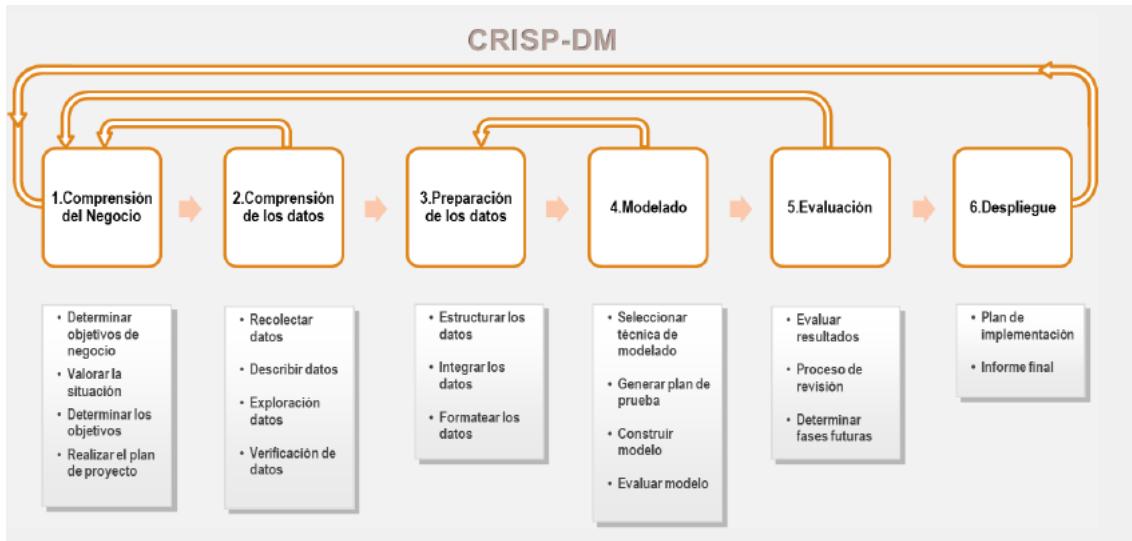
3.- Desarrollo

En la creación de nuestro proyecto hemos apostado por una metodología de trabajo **Agile**, así como la utilización de frameworks de trabajo acordes con esa filosofía o estilo de trabajo. Además, para la parte de desarrollo nos basamos en herramientas colaborativas que nos permiten una mejor coordinación y optimización del trabajo en equipo. Detallamos a continuación la metodología seguida,

3.1 Metodología

El objeto de este trabajo es la comprensión de las magnitudes y variables que arrojan datos con relación a la calidad del aire y contaminación atmosférica de la comunidad de Madrid, incluyendo la ciudad de Madrid, durante el periodo de confinamiento vivido en España a raíz del Decreto que sometió a confinamiento a todo el país, visualización los cambios en los datos recogidos por la red de estaciones que el Ayuntamiento tiene desplegadas en la capital.

Todo ello siguiendo la metodología **CRISP-DM** utilizada en proyectos de data science. Por ello se ha dedicado gran parte del tiempo en el estudio del arte y la comprensión de los datos, el preprocesado de los datos, su análisis, el análisis dependencias y la tendencia, además de varios ajustes realizados en el entorno de trabajo para las diversas etapas en las que consistirá el proyecto.



Para ello, lo primero fue documentarnos sobre las principales variables de la contaminación atmosférica, las técnicas de análisis de series temporales y las distintas librerías que ofrece Python para su manejo.

A continuación, exploración de toda la información disponible en los sitios web del Ayuntamiento y la Comunidad de Madrid, obteniendo la información al respecto de la calidad del aire en Madrid y del tiempo.

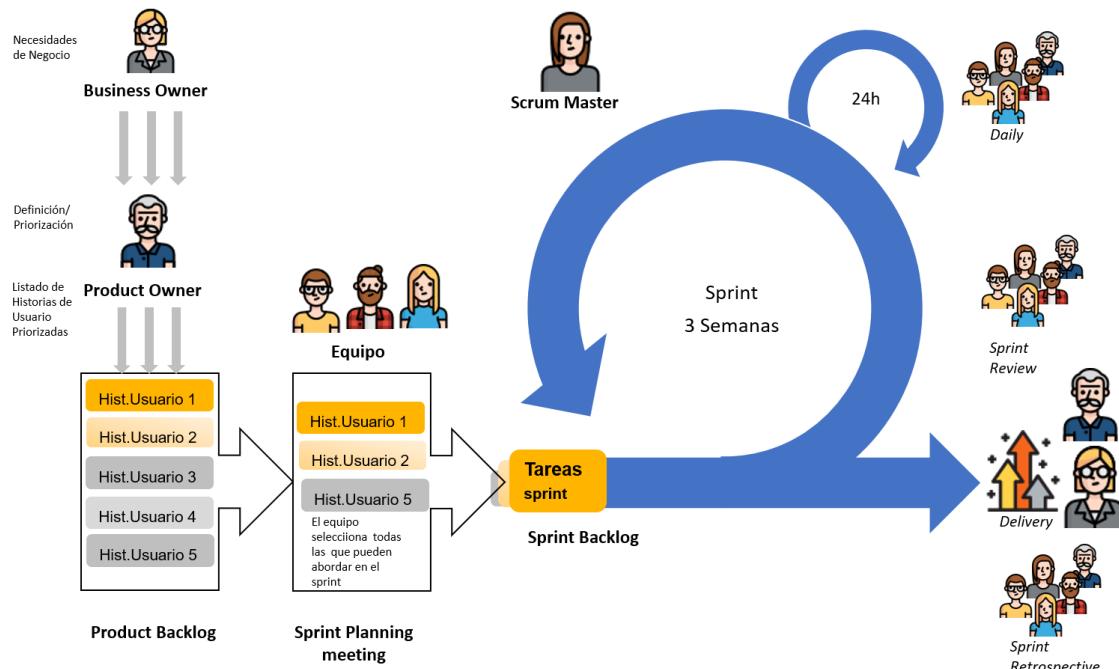
Mediante el uso de tablas y gráficos, se muestran distintos agregados de datos según las etapas del confinamiento que abarcan desde Enero a Mayo, usando la media y la mediana como estimadores.

Además, el equipo de trabajo ha utilizado la metodología Agile.

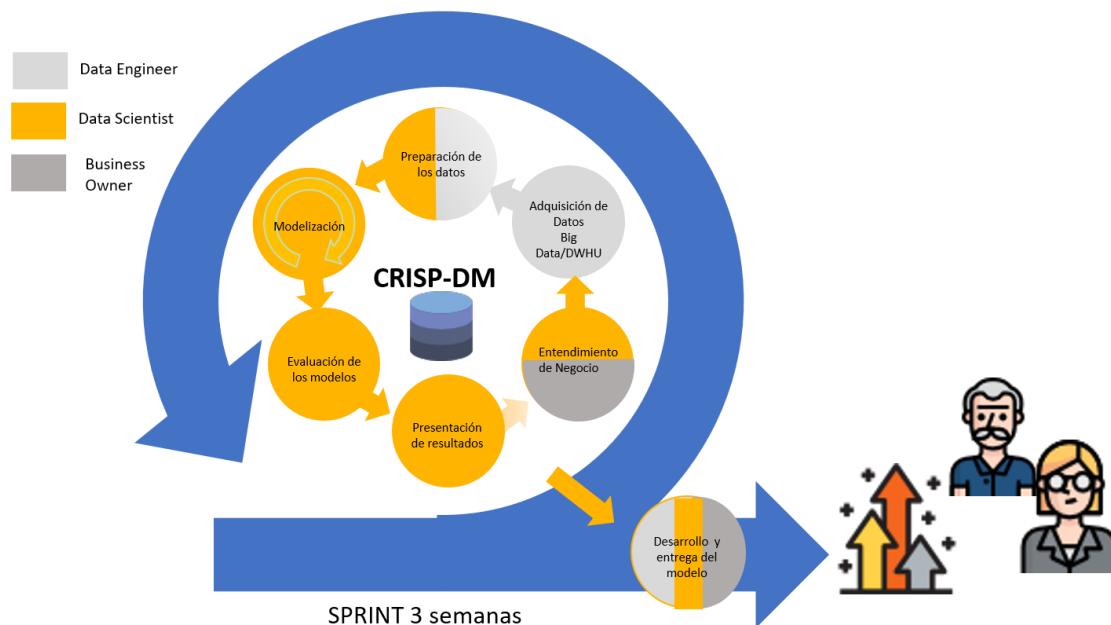
Agile es más que una metodología; ser Agile es tener una visión consensuada, que se centra en cómo las personas trabajan, se relacionan y cómo consiguen resultados. Por ello, ha sido la metodología empleada por el equipo.

El uso de ambas metodologías nos permite definir dentro del proyecto una nueva metodología: **Agile Data Science** para trabajar con ambas metodologías en ciclos. El esquema de ambas metodologías superpuestas sería el siguiente:

Metodología Agile



+ CRISP-DM = AGILE DATA SCIENCE



3.2 Roles

La realización del proyecto se realiza otorgando una serie de Roles a cada uno de los integrantes, que nos permiten establecer una distribución de las tareas acorde con las habilidades individuales que caracterizan a todos los participantes.



ANALISTA DE DATOS

Susana Vara, analista de negocio en Orange España. Actualmente trabajo en la tribu de Jazztel como Business Analyst, Estudié Estadística en la Universidad de Valladolid y eso me dio una gran versatilidad entre la informática y las matemáticas, utilizando muchos lenguajes de programación y programas informáticos dentro de la universidad. Disfruto trabajando en equipo y por eso la metodología agile me encantó desde el primer momento e incluso en ocasiones he llegado a desarrollar la función de scrum master dentro del squad.



INGENIERO DE DATOS

Raúl Hervás: Tras pasar por varios departamentos de Sistemas y Redes, pasé a dedicarme al desarrollo de aplicaciones enfocadas principalmente a la web. Actualmente soy desarrollador en Grupo CTO en el cual me ocupo de distintas aplicaciones internas y en el desarrollo del campus Virtual para los alumnos de diversas oposiciones de Medicina, Enfermería, Hacienda...



ANALISTA DE NEGOCIO

Carmen Reina: Soy Matemática, Científica de datos y experta en

Inteligencia artificial. He desarrollado a lo largo de mi carrera profesional distintas actividades vinculadas al tratamiento y la generación de conocimiento en distintas áreas. Actualmente soy Head of Data Culture en Orange, donde intento trasmisir mi pasión en las tecnologías. Me encargo de la definición de estrategia formativa y cultural para el Área de Datos.



Alfonso Gallardo: Soy Analytics Specialist en [Avanade Spain](#) empresa del grupo Microsoft. Con más de 6 años de experiencia en el mundo de los datos, he desarrollado proyectos para empresas del sector financiero y otras empresas del sector público.

CIENTIFICO DE DATOS



ARQUITECTO DE DATOS

Walter Ronceros: He sido Service Manager en Orange en el área de la operación durante los últimos 3 años y medio pero acabo de cambiar de aires y y he "aireizado" en el mundo Cloud. Actualmente soy Cloud Enginner y pertenezco al Centro de Excelencia de Cloud en Orange donde se desarrollan e implantan todos los proyectos híbridos y Cloud Native de la compañía en cuanto a la infraestructura cloud se refiere.

3.3. Tecnologías empleadas

Las tecnologías utilizadas para el desarrollo del proyecto son las siguientes:



Figura X. Tecnologías usadas

3.3.1 Python

El lenguaje de programación en el que hemos desarrollado el código y algoritmos es Python en su última versión actual que es la [Python 3.8.2](#) publicada el 13 de mayo de 2020.

3.3.2 Anaconda Navigator

Anaconda es una Suite de código abierto que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la Ciencia de datos con Python.

La versión del navegador es la 3.0.1

3.3.3 Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

La versión del servidor es 19.03.8

3.3.4 Jupyter Notebook

Jupyter Notebook es una aplicación web de código abierto que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más.

La versión de Jupyter Notebook es 6.0.3

Con Jupyter hemos realizado todo el análisis del dataset final, transformación de los datos, descripción y modelado.

3.3.5 Spyder

Spyder es un poderoso entorno de desarrollo escrito en Python, para Python, ofrece integración integrada con muchos paquetes científicos populares, incluidos NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy y más.

Con Spyder hemos realizado el proceso de generación de los datasets para posteriormente tratar y analizar en Jupyter.

3.3.6 Elasticsearch

Es un motor de analítica y análisis distribuido y open source para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y desestructurados. Elasticsearch está desarrollado en Apache Lucene y fue presentado por primera vez en 2010 por Elasticsearch N.V. (ahora conocido como Elastic). Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad, Elasticsearch es el componente principal del Elastic Stack, un conjunto de herramientas open source para la ingesta, el enriquecimiento, el almacenamiento, el análisis y la visualización de datos. Comúnmente referido como el ELK Stack (por Elasticsearch, Logstash y Kibana), el Elastic Stack ahora incluye una gran colección de agentes de envío conocidos como Beats para enviar los datos a Elasticsearch.

3.3.7 Databricks

Databricks es una empresa fundada por los creadores originales de Apache Spark. Databricks surgió del proyecto AMPLab en la Universidad de California, Berkeley, que estuvo involucrado en la fabricación de Apache Spark, un marco de cómputo distribuido de código abierto construido sobre Scala.

3.4 Entorno Colaborativo

Hemos creado durante el proyecto un entorno colaborativo para poder trabajar conjuntamente en los scripts y documentos. Para ello, hemos usado herramientas sólo herramientas Open Source.

3.4.1 Google Colab

Para el desarrollo del proyecto se ha empleado el lenguaje de programación **Python** para la conexión a las APIs y tratamiento de los datos para obtener los databases a estudio, así como para la elaboración de los modelos predictivos.

Hemos utilizado varias librerías para el tratamiento y modelado de los datos. Estas son: Pandas, matplotlib, scikit-learn (incluyendo varias sublibrerías), seaborn, Prophet,xgboost.

Para trabajar conjuntamente en estos scripts, hemos usado a lo largo del proyecto [**Google Colab**](#).

En Colab hemos compartimos un notebook para que todos los miembros del equipo podemos ver y trabajar en el desarrollo del proyecto. Mostramos una imagen del notebook generado.

```

[ ] # Cargamos las librerías que vamos a usar.
import pandas as pd
import datetime
from fbprophet import Prophet
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline
from scipy.stats import norm
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns

[ ] /usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.

[ ] import pandas as pd
from google.colab import files

uploaded = files.upload()

[ ] [Clear errors] Ningún archivo seleccionado Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving datosdefinitivos.csv to datosdefinitivos.csv

[ ] # Carga dataframe
datos_original = pd.read_csv('datosdefinitivos.csv',sep=';')

```

Ilustración 1. Notebook en Google Colab.

3.4.2 GitHub

Como herramienta de compartición de trabajo, utilizamos [GitHub](#), plataforma de desarrollo colaborativo de software para alojar el proyecto. Los miembros del equipo que no disponían de usuario en esta aplicación se han creado uno.

Creamos en [GitHub](#) un proyecto nuevo donde hemos compartido las versiones de cada documento de los que componen el proyecto.

<https://github.com/Big-Data-Equipo-7/Proyecto>

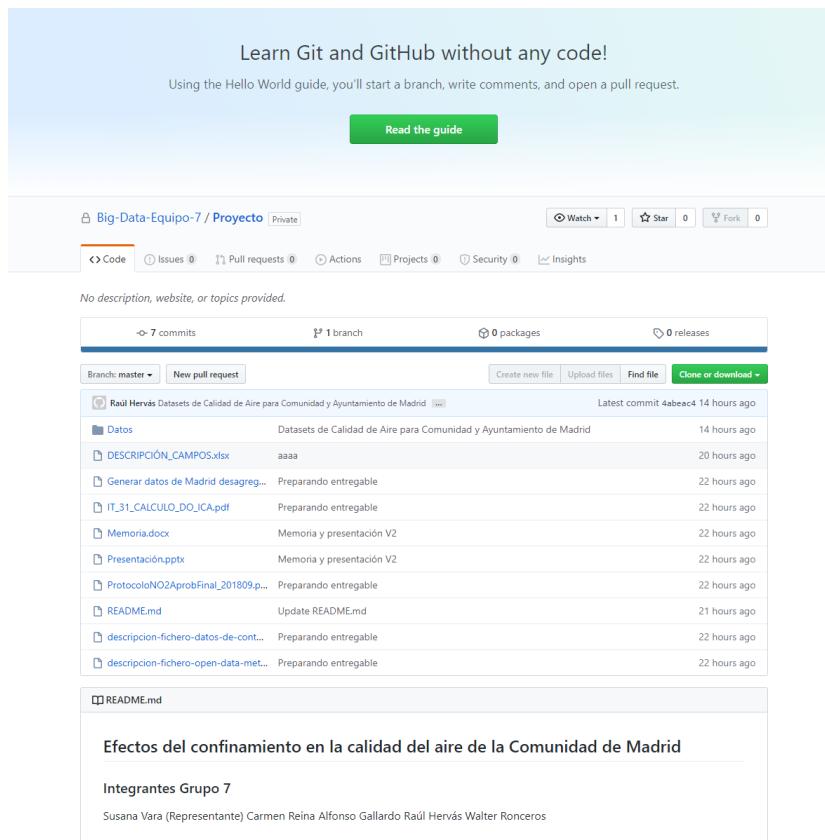


Ilustración 2. Proyecto en GitHub.

3.4.2 Trello

Como soporte a la metodología Agile, la coordinación dentro del equipo la hemos gestionado en [Trello](#), software de administración de proyectos con interfaz web y con cliente para iOS y Android con el fin de organizar proyectos.

Trello nos ha permitido definir las tareas asociadas a cada integrante del equipo así como crear las Historias de Usuario y tareas que íbamos a desarrollar en el proyecto. De esta forma hemos tenido una visión clara de los trabajos a desarrollar así como la responsabilidad de cada miembro del equipo. También hemos usado Trello como gestor documental para compartir la documentación relacionada con el conocimiento del problema que estamos tratando de la contaminación del aire, enlaces a open data de la Comunidad y del Ayuntamiento de Madrid y otra documentación de interés.

Mostramos abajo una imagen del proyecto de Trello generado.

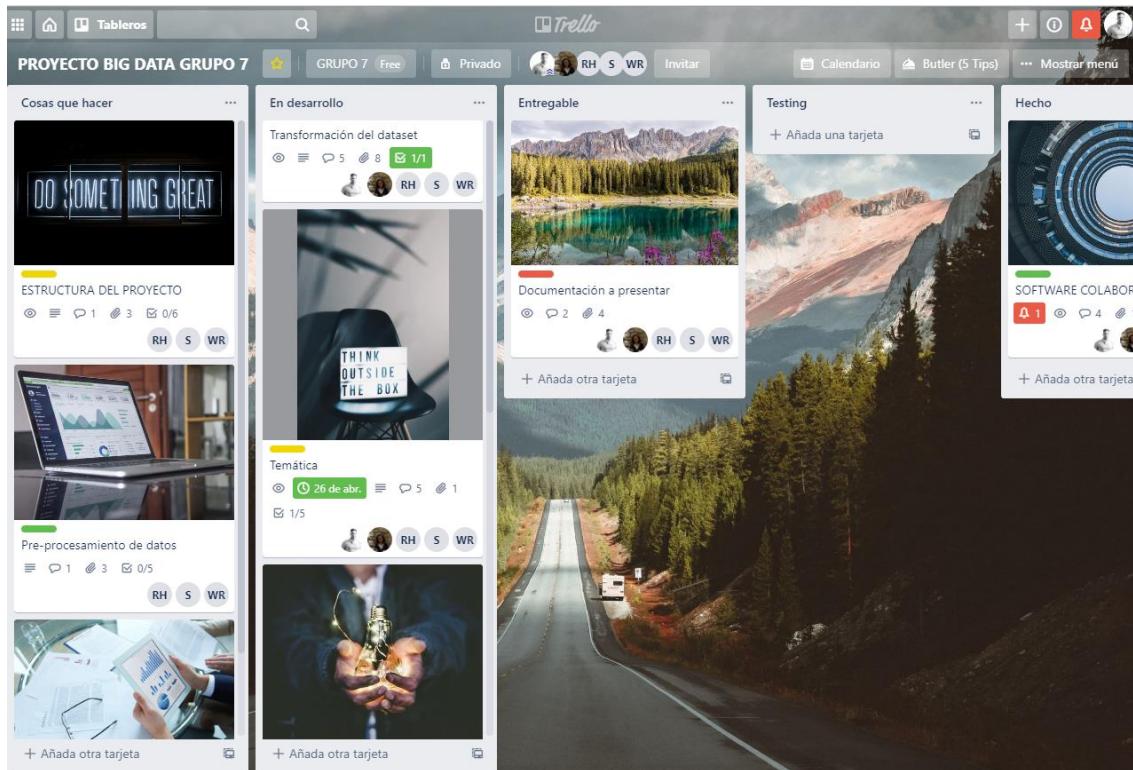


Ilustración 3. Entorno de trabajo en Trello

4.- Fases del proyecto

El proyecto lo hemos dividido en fases siguiendo la metodología Crisp-DM arriba indicada. A continuación describimos cada fase, así como las tareas emprendidas en ellas y sus resultados.

4.1 Comprensión de Negocio

En esta fase, hemos identificado el objetivo a alcanzar, identificando todos los aspectos relevantes para ello. Hemos procedido a la búsqueda de información relevante de la Comunidad de Madrid, Ayuntamiento de Madrid y estudios sobre la contaminación atmosférica (ver apartado de bibliografía).

A continuación procedemos a enumerar algunos aspectos que nos han parecido relevantes para el entendimiento y resolución de la problemática estudiada y de cómo se está abordando a día de hoy por parte de la Comunidad de Madrid y el Ayuntamiento de Madrid.

4.1.1 Sustancias contaminantes

Hay diversas maneras de clasificar las sustancias contaminantes en función de distintos criterios.

En cuanto a su procedencia, se agrupan de la siguiente forma:

1. **Primarios:** proceden directamente de las fuentes de emisión.
 - Gaseosos
 - Dióxido de azufre (SO₂)
 - Monóxido de carbono (CO)
 - Óxidos de nitrógeno (NO_x)
 - Hidrocarburos (HC)
 - Dióxido de carbono (CO₂)
 - No gaseosos
 - Partículas: su procedencia y composición es muy variada.
 - Restos de combustión de fuel, gas-oil o alquitranes
 - Erupciones volcánicas
 - Incendios
 - Intrusiones de material particulado
 - Incineraciones no depuradas de basuras
 - Metales pesados: plomo, cadmio, mercurio, etc.
2. **Secundarios:** se originan en la atmósfera como consecuencia de reacciones químicas que transforman los contaminantes primarios
 - Ozono (O₃)
 - Trióxido de azufre (SO₃)
 - Ácido sulfúrico (H₂SO₄)
 - Dióxido de nitrógeno (NO₂)
 - Ácido nítrico (HNO₃)
 -

Los contaminantes causan distintos efectos negativos sobre el medio y los seres vivos. Además, algunos contaminantes secundarios son responsables de importantes alteraciones en la calidad del aire.

4.1.2 Estaciones de monitorización de la calidad del aire del Ayuntamiento de Madrid y de la Comunidad de Madrid

El Sistema de Vigilancia está formado por **37 estaciones remotas**, en el ayuntamiento de Madrid y **24 en la comunidad de Madrid**. Automáticas que recogen la información básica para la vigilancia atmosférica. Poseen los analizadores necesarios para la medida correcta de los niveles de gases y de partículas.

Estaciones de la Red de Vigilancia



Plaza de España



Escuelas Aguirre



Ramón y Cajal



Arturo Soria



Villaverde



Farolillo



Casa de Campo



Barajas Pueblo



Plaza del Carmen



Moratalaz



Cuatro Caminos



Barrio del Pilar



Vallecas



Méndez Álvaro



Castellana



Retiro



Plaza de Castilla



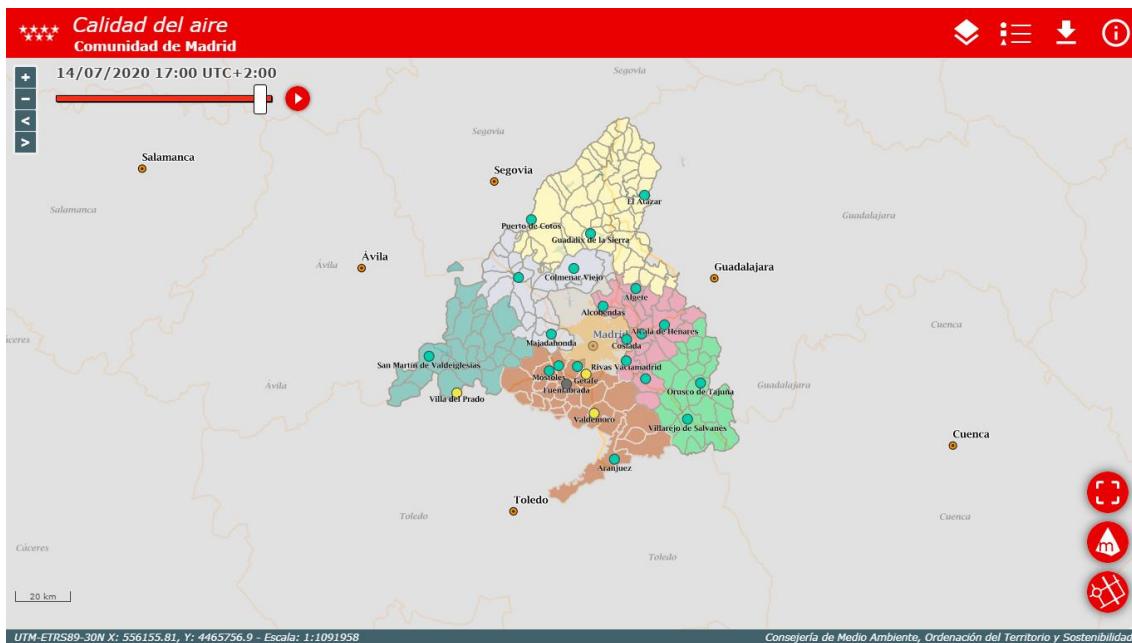
Ensanche de Vallecas

Las estaciones remotas del ayuntamiento de Madrid son de varios tipos:

- **Urbanas de fondo:** Representativas de la exposición de la población urbana en general.
- **De tráfico:** Situadas de tal manera que su nivel de contaminación está influido principalmente por las emisiones procedentes de una calle o carretera próxima, pero se ha de evitar que se midan microambientes muy pequeños en sus proximidades.
- **Suburbanas:** Están situadas a las afueras de la ciudad, en los lugares donde se encuentran los mayores niveles de ozono.

En la imagen se ven las estaciones de la comunidad de Madrid, instaladas en los municipios de :

ALCALÁ DE HENARES ,ALCOBENDAS ,ALCORCÓN ,ALGETE ,ARANJUEZ ,ARGANDA DEL REY ,EL ATAZAR ,COLMENAR VIEJO ,COLLADO VILLALBA ,COSLADA ,FUENLABRADA ,GETAFE ,GUADALIX DE LA SIERRA ,LEGANÉS ,MAJADAHONDA ,MÓSTOLES ,ORUSCO DE TAJUÑA ,PUERTO DE COTOS ,RIVAS-VACIAMADRID ,SAN MARTÍN DE VALDEIGLESIAS ,TORREJÓN DE ARDOZ ,VALDEMORO ,VILLA DEL PRADO ,VILLAREJO DE SALVANÉS .



4.1.3 Registro de Datos, Interprete de Ficheros de datos horarios, diarios y tiempo real

Los datos se proporcionan en tres formatos distintos, .txt, .csv, .xml

Datos horarios y en tiempo real:

Cada registro está estructurado de la siguiente forma:

1. Antes de octubre de 2017:

2807900401380217070100005V00004V00004V00004V00004V...

Los caracteres se corresponden con lo siguiente:

PROVINCIA	MUNICIPIO	ESTACIÓN	MAGNITUD	TÉCNICA	PERÍODO ANÁLISIS	AÑO	MES	DÍA	DATO	CÓDIGO DE VALIDACIÓN
28	079	004	01	38	02	17	07	01	00005	V

2. Desde octubre de 2017:

28,079,004,01,38,02,2019,01,01,00023,V,00045,V,00028,V,00037,V,...

Los caracteres se corresponden con lo siguiente:

PROVINCIA	MUNICIPIO	ESTACIÓN	MAGNITUD	TÉCNICA	PERÍODO ANÁLISIS	AÑO	MES	DÍA	DATO	CÓDIGO DE VALIDACIÓN
28	079	004	01	38	02	2019	01	01	00023	V

Para los archivos de datos horarios y en tiempo real, el código del periodo de análisis es 02 y las columnas correspondientes a los datos son 24, una por hora, cada una de ellas con su correspondiente código de validación.

Únicamente son válidos los datos marcados con Código de Validación "V".

Datos diarios:

Cada registro está estructurado de la siguiente forma:

28,079,004,01,38,02,2019,01,00008,V,00009,V,00008,V,00007,V,...

Los caracteres se corresponden con lo siguiente:

PROVINCIA	MUNICIPIO	ESTACION	MAGNITUD	PUNTO_MUESTREO	ANO	MES	DIA	H01	V01	H02	V02
28	79	4	1	28079004_1_38	2019	1	1	23	V	17	V

El campo punto de muestreo incluye el código de la estación completo (provincia, municipio y estación) más la magnitud y la técnica de muestreo.

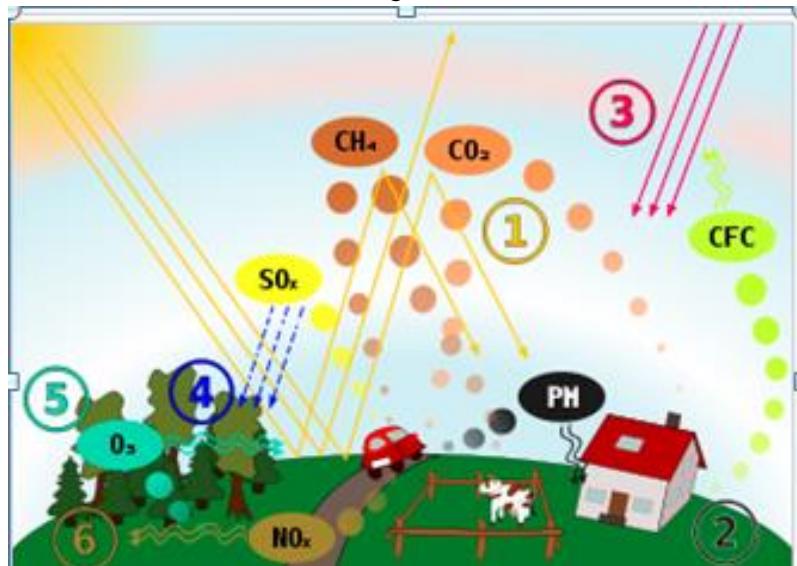
H01 corresponde al dato de la 1 de la mañana de ese día, V01 es el código de validación, H02 al de las 2 de la mañana, V02 y así sucesivamente. D01 corresponde al dato del primer día del mes, D02 al del segundo día y así sucesivamente.

4.1.4 Índice de calidad del aire

La calidad del aire a nivel mundial se mide mediante un índice denominado **AQI** (Air Quality Index), en español es **ICA (Índice de la Calidad del Aire)**.

Este índice es el máximo de los valores equivalentes de 5 contaminantes: SO2 (Dióxido de Azúfre), NO2 (Dióxido de Nitrógeno), CO (Monóxido de Carbono) , O3(Ozono), PM10 y PM25 (partículas) en todas las estaciones de medida de un municipio o región.

En el dibujo podemos ver las causas y efectos de la contaminación del aire: (1) efecto invernadero, (2) contaminación por partículas, (3) aumento de la radiación UV, (4) SO₂ dióxido de azufre (lluvia ácida), (5) aumento de la concentración de ozono a nivel del suelo, (6) aumento de los niveles de óxidos de nitrógeno.



Mediante diferentes colores proporciona información rápida y comprensible sobre el grado de contaminación atmosférica de una determinada zona. Cada color está definido por un adjetivo que expresa la mejor o peor calidad del aire. De esta forma se puede relacionar fácilmente la calidad del aire que respira con potenciales repercusiones en su salud.

Acerca de los niveles de calidad del aire

ICA	Calidad del Aire	Proteja su Salud
0 - 50	Buena	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este intervalo.
51 - 100	Moderada	Las personas extraordinariamente sensibles deben considerar limitación de los esfuerzos físicos excesivos y prolongados al aire libre.
101-150	Dañina a la Salud de los Grupos Sensitivos	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos físicos excesivos y prolongados al aire libre.
151-200	Dañina a la Salud	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos excesivos prolongados al aire libre; las demás personas, especialmente los niños, deben limitar los esfuerzos físicos excesivos y prolongados al aire libre.
201-300	Muy Dañina a la Salud	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar todos los esfuerzos excesivos al aire libre; las demás personas, especialmente los niños, deben limitar los esfuerzos físicos excesivos al aire libre.
300+	Arriesgado	

4.1.4.1 Variable Dióxido de azufre (SO₂)

El dióxido de azufre (SO₂) es un gas incoloro, no inflamable. Posee un olor fuerte e irritante en altas concentraciones. Se origina por la combustión de

carburantes con cierto contenido en azufre (carbón, fuel) y la fundición de minerales ricos en sulfatos. Se genera principalmente por la industria (incluyendo las termoeléctricas), seguido de los vehículos a motor.

4.1.4.2. Dióxido de Nitrógeno (NO₂)

El dióxido de nitrógeno (NO₂) es un contaminante indicador de actividades de transporte, especialmente el tráfico rodado. Lo emiten directamente los vehículos, especialmente los diesel (emisiones directas o «primarias»), pero se produce también en la atmósfera a partir de las emisiones de monóxido de nitrógeno (NO) de los vehículos; por un proceso químico, dicho gas se transforma en NO₂ (contaminante «secundario»).

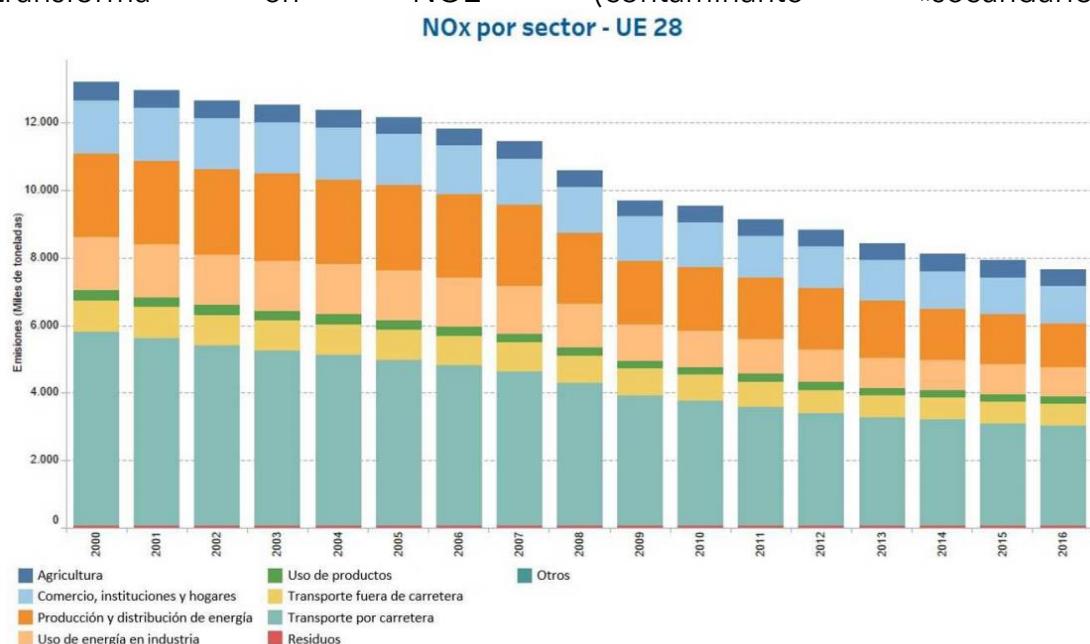


Ilustración 4. Evolución emisiones dióxidos de nitrógeno UE28. (Fuente: Agencia Europea del Medio Ambiente)

Además de contribuir en la formación de ozono, se relaciona al NO₂ con efectos nocivos sobre el sistema respiratorio. Los dióxidos de nitrógeno reaccionan con el amoniaco, con la humedad y otros compuestos para formar pequeñas partículas. Estas pequeñas partículas pueden penetrar profundamente en las partes sensibles de los pulmones.

La evidencia científica vincula exposiciones cortas a NO₂ (desde 30 minutos hasta 24 horas) con efectos adversos respiratorios, incluida la inflamación de las vías respiratorias en personas sanas y el aumento de los síntomas en personas que padecen de asma. Los estudios también muestran que existe conexión entre la exposición a corto plazo a este contaminante y el aumento de visitas a las emergencias hospitalarias por problemas respiratorios.

El NO₂ también es uno de los causantes de la conocida lluvia ácida, ya que al reaccionar con el vapor de agua produce ácido nítrico. Los efectos sobre la agricultura, la ganadería, los bosques, los suelos y las aguas son muy graves.

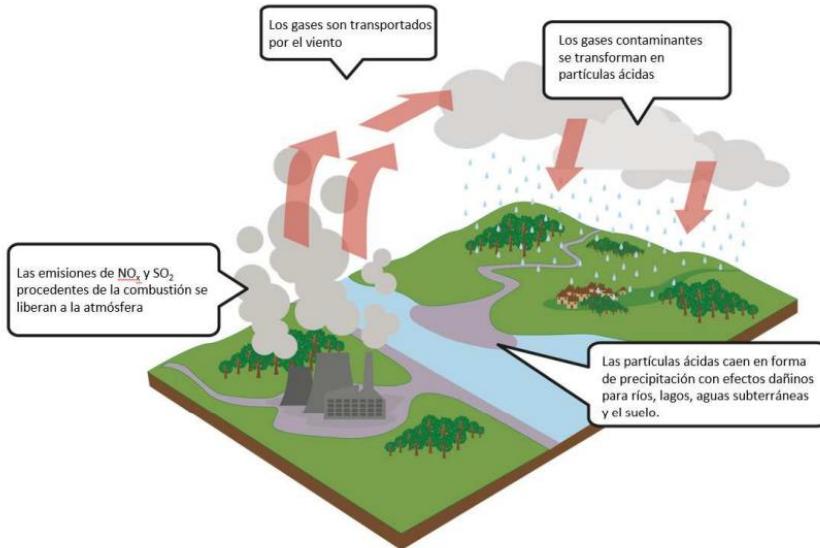


Ilustración 5. Esquema de formación de lluvia ácida

En este sentido, llevar a cabo acciones encaminadas a reducir el impacto de la contaminación atmosférica ha sido un objetivo primordial.

Los distintos estados miembros de la UE han ido incorporando progresivamente a su legislación distintas medidas para alcanzar los objetivos planteados en la Directiva 2008/50/CE. Algunas de estas medidas consisten en protocolos de actuación en materia de tráfico ante episodios de alta contaminación ya que, como se muestra en la Figura 1, el tráfico por carretera es uno de los principales orígenes del NO₂. Madrid, por ejemplo, ha puesto en marcha un protocolo de actuación en episodios de alta contaminación. Esta ha sido una de las motivaciones para realizar el presente trabajo: tratar de estudiar si, a través de la minería de datos podemos predecir si un día se dará o no un episodio de alta contaminación.

4.1.4.3 Partículas en suspensión (PM10)

El material particulado es una mezcla compleja de componentes con características químicas y físicas diversas, formadas a partir de otros contaminantes primarios e, incluso, a partir de elementos naturales.

En las ciudades europeas, este material se genera en procesos de combustión provenientes tanto de los sistemas de calefacción de edificios como de las emisiones generadas por el tráfico rodado, con una especial

importancia en los motores de ciclo diesel con tecnologías de motor anteriores al año 2000.

Además, en el caso de España, por su situación geográfica, se pueden encontrar aportes de origen natural como pueden ser las procedentes del desierto del Sáhara. El término PM10 se refiere a partículas en suspensión con un diámetro aerodinámico de hasta 10 μm , comprendiendo las fracciones fina y gruesa, y PM2.5 se refiere a partículas en suspensión con un diámetro aerodinámico de hasta 2.5 μm .

4.1.4.4. Monóxido de Carbono (CO)

El monóxido de carbono es un contaminante primario indicador del tráfico rodado. Es un gas incoloro, inodoro e insípido.

Su presencia se ha reducido de manera continua en los últimos años debido fundamentalmente a los cambios tecnológicos en los vehículos de motor que son los principales emisores de este contaminante.

4.1.4.5. Ozono (O₃)

El ozono es un contaminante secundario que se forma a partir de una serie de contaminantes precursores cuando encuentran un nivel de insolación suficiente. Las moléculas de este gas azulado y picante están formadas por tres átomos de oxígeno.

Presenta dos propiedades que marcan sus interacciones con la vida de nuestro planeta: su fuerte absorción de la radiación ultravioleta y su gran poder oxidante.

La primera hace que su presencia en la estratosfera sea imprescindible como filtro para evitar que lleguen a la superficie del planeta altos niveles de radiación ultravioleta que resultarían catastróficos para todos los seres vivos. Por eso existen tantas campañas y esfuerzos para evitar el deterioro de la conocida «capa de ozono».

Sin embargo, la segunda propiedad -su alto poder oxidante-, lo hace muy peligroso cuando aparece en la troposfera porque, en determinadas concentraciones, puede producir daños en nuestra salud, en la vegetación y en los materiales.

4.1.5 Adaptación a la Directiva 2008/50/CE en España

Centrándonos en España, el Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire, establece umbrales de alerta para algunos agentes contaminantes, entre ellos el dióxido de nitrógeno. Se define el umbral de alerta como "el nivel a partir del cual una exposición de breve

duración supone un riesgo para la salud humana, que afecta al conjunto de la población y que requiere la adopción de medidas inmediatas."

El valor del umbral de alerta para el dióxido de nitrógeno está establecido en 400 microgramos/m³ durante tres horas consecutivas en lugares representativos de la calidad del aire, en un área de al menos 100 km² o en una zona o aglomeración entera, si esta última superficie es menor. El citado Real Decreto establece asimismo un valor límite horario para la protección de la salud de dióxido de nitrógeno de 200 microgramos/m³ (nivel de aviso) que no debe superarse más de 18 horas al año en ninguna de las estaciones de la red.

4.1.6 Protocolo de Contaminación en Madrid

El Ayuntamiento de Madrid, para llevar a cabo el control de la calidad del aire de la ciudad, dispone del Sistema de Vigilancia, Predicción e Información de la Calidad del Aire que permite conocer, de forma continua y en tiempo real, las concentraciones de contaminantes, con el principal objetivo de proteger la salud de la población y reducir al máximo las situaciones de riesgo.

Las elevadas concentraciones son originadas fundamentalmente por las emisiones del tráfico, y tienen lugar en situaciones con condiciones meteorológicas especialmente adversas, que requieren la ejecución de medidas para reducir los niveles de contaminación y la duración de los episodios, y evitar que llegue a superarse el valor límite horario y que se llegue a alcanzar el umbral de alerta.

El Ayuntamiento de Madrid ha establecido una división en zonas de tal manera que las situaciones de alerta puedan declararse en áreas más reducidas con alta densidad de población. Igualmente se definen unos niveles de aviso que permitan, en el caso de registrarse concentraciones elevadas de dióxido de nitrógeno, la puesta en marcha de mecanismos de información adicionales, que sirvan tanto para proteger la salud de los ciudadanos como para sensibilizar a la opinión pública, recabar su colaboración para la reducción de la contaminación y, en función de los niveles alcanzados y la duración del episodio, llevar a cabo medidas de restricción de tráfico en la ciudad y sus accesos para reducir los niveles de contaminación y evitar que se alcance la situación de alerta. La ciudad de Madrid, a efectos de aplicación del Protocolo de medidas a adoptar durante episodios de alta contaminación, se ha dividido en cinco zonas. Cada una de las estaciones de medición del aire se encuentra enmarcada en alguna de estas zonas siendo la distribución la que se indica a continuación:

Zona	Estaciones
1 (Interior M30)	7 de tráfico (Escuelas Aguirre, Castellana, Plaza de Castilla, Ramón y Cajal, Cuatro Caminos, Plaza de España y Barrio del Pilar) + 3 de fondo (Plaza del Carmen, Méndez Álvaro y Retiro)
2 (Sureste)	1 de tráfico (Moratalaz) + 2 de fondo (Vallecas y Ensanche de Vallecas)
3 (Noreste)	5 de fondo (Arturo Soria, Sanchinarro, Urbanización Embajada, Barajas pueblo y Tres Olivos) + 1 suburbana (Juan Carlos I)
4 (Noroeste)	2 suburbanas (El Pardo y Casa de Campo)
5 (Suroeste)	1 de tráfico (Fernández Ladreda) + 2 de fondo (Farolillo y Villaverde)

Ilustración 6. Estaciones calidad aire por zona. (Fuente: Ayto. Madrid - Protocolo de actuación para episodios de contaminación por dióxido de nitrógeno)

Las zonas se han definido atendiendo a los siguientes criterios:

- La distribución de la población.
- La tipología y distribución de estaciones del sistema de vigilancia de la calidad del aire.
- El viario de tráfico, para facilitar la implantación de posibles actuaciones de restricción de este.

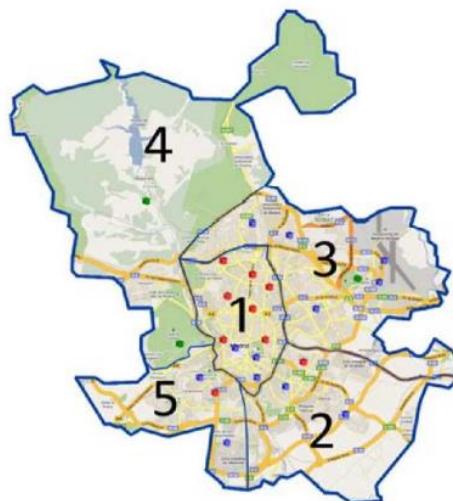


Ilustración 7. Delimitación de zonas a efectos de aplicación del protocolo de actuación para episodios de contaminación por dióxido de nitrógeno. (Fuente: Ayto. Madrid)

- **Zona 1:** área comprendida en el interior de la M30.
- **Zona 2:** área delimitada por la Avda. de Andalucía, Calle 30, la autovía M23 continuando por la R3 y hasta el límite del término municipal de Madrid.
- **Zona 3:** área delimitada por la autovía M23 y la continuación de la R3, Calle 30 hasta la M40 en la zona oeste y desde allí limita al norte con

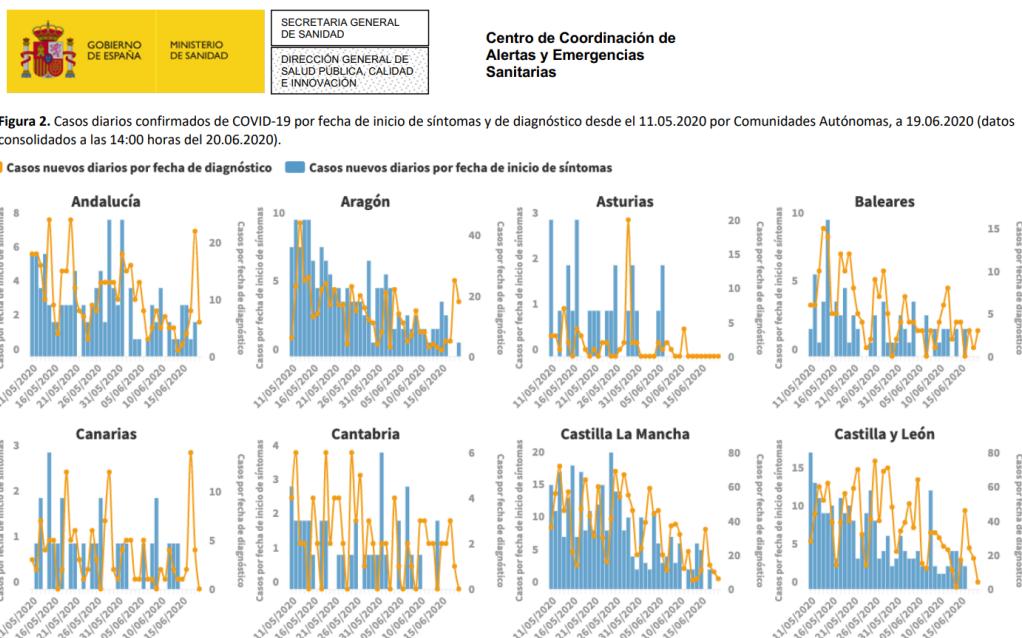
la M40 hasta el límite del término municipal de Madrid. Esta zona incluye parte del Aeropuerto de Barajas.

- **Zona 4:** área delimitada por el contorno del límite del municipio de Madrid por el norte, la M40 norte, Calle 30 hasta la A5 y el límite del municipio.
- **Zona 5:** área delimitada por el contorno sur de la Casa de Campo, Calle 30, Avda. de Andalucía y el término municipal de Madrid. Se establecen tres niveles de actuación en función de las concentraciones de dióxido de nitrógeno que se registren en las zonas que se han definido:

- Nivel de preaviso: cuando en dos estaciones cualesquiera de una misma zona se superan los 180 microgramos/m³ durante dos horas consecutivas.
- Nivel de aviso: cuando en dos estaciones cualesquiera de una misma zona se superan los 200 microgramos/m³ durante dos horas consecutivas.
- Nivel de alerta: cuando en tres estaciones cualesquiera de una misma zona (o dos si se trata de la zona 4) se superan los 400 microgramos/m³ durante tres horas consecutivas.

4.1.7 Cronología del COVID en España

El viernes 31 de enero de 2020, mientras que en todo el planeta se contabilizaban menos de 10 000 infectados confirmados, el Ministerio de Sanidad comunicó el primer caso positivo de COVID-19 en España, un turista alemán que se encontraba hospitalizado y aislado en el Hospital de Nuestra Señora de Guadalupe, en la isla de La Gomera. Este primer paciente fue una de las cinco personas que se encontraban en observación al haber estado en contacto con un positivo en Alemania.



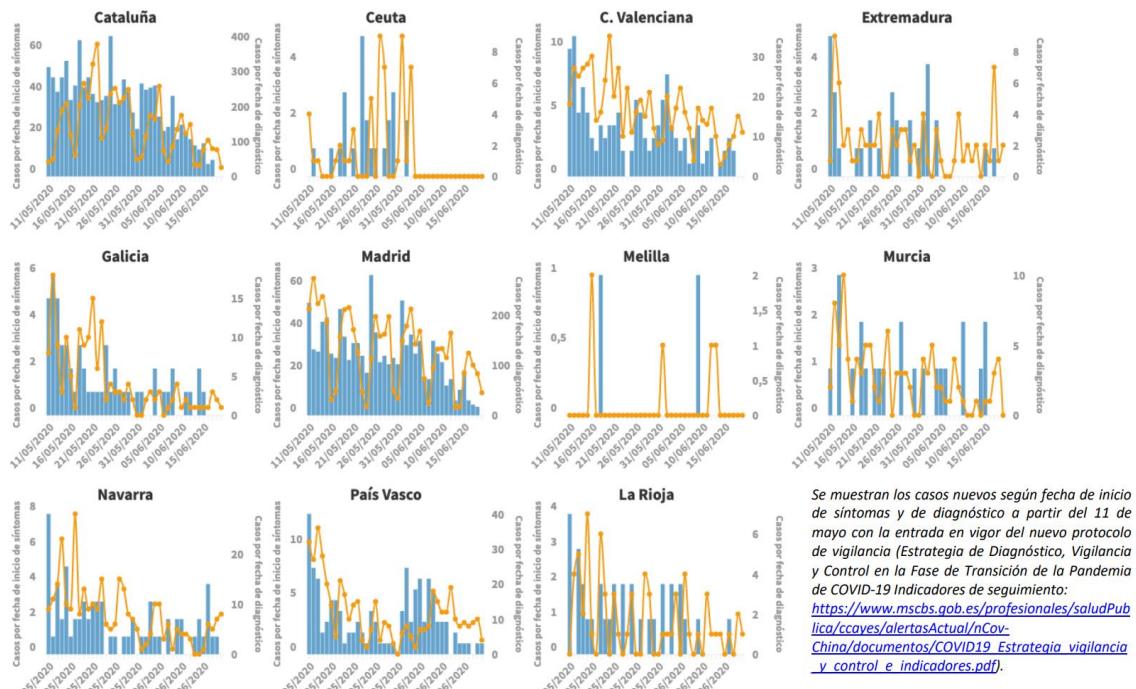


Ilustración 8. Casos diarios confirmados de COVID-19 por fecha de inicio de síntomas y de diagnóstico desde el 11/05/2020 por Comunidades Autónomas, a 19/06/2020

El segundo caso fue diagnosticado el 10 de febrero en un paciente británico residente en Palma de Mallorca. Esta persona contrajo la enfermedad al estar en contacto con un compatriota en los Alpes, el cual se infectó en un viaje a Singapur. Esta persona fue ingresada y aislada junto con su cónyuge y sus dos hijas en el Hospital de Son Espases.²⁰

El 12 de febrero, el mayor congreso tecnológico del mundo, el Mobile World Congress de Barcelona, fue cancelado a pesar de que las autoridades sanitarias insistían en que no existía ningún riesgo. La decisión se tomó tras confirmar algunas de las mayores empresas de tecnología del planeta que suspendían su presencia en el congreso, entre las que se encontraban LG, Facebook, Sony o Vodafone, por el miedo al contagio a gran escala de los asistentes. Esta cancelación supuso al sector hotelero de Barcelona un duro golpe económico.

Durante los días 24 y 25 de febrero, se reportaron varios casos positivos repartidos por España de la enfermedad por coronavirus provenientes de la epidemia del norte de Italia: un médico y su mujer, procedentes de Lombardía que se encontraban de vacaciones en Tenerife y que supuso la cuarentena de los 700 huéspedes del hotel en el que se encontraban una mujer italiana residente en Barcelona y que visitó Bérgamo y Milán en fechas recientes en Villarreal, un varón que había estado recientemente de viaje en Milán y un joven en Madrid que había estado de viaje por el norte del país transalpino.

En Sevilla, el 26 de febrero, se detecta el primer infectado confirmado en territorio andaluz.

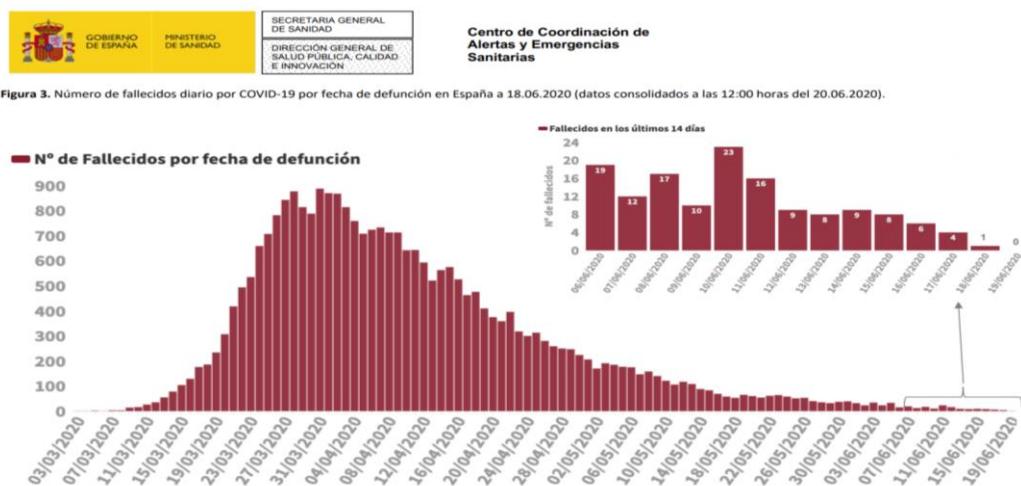


Ilustración 9. Número de fallecidos diario por COVID-19 por fecha de defunción en España a 18/06/2020

El 14 de marzo, cuando en España se contaban alrededor de 6.000 casos y de 200 muertos, el Consejo de Ministros **declaró el estado de alarma** en todo el territorio nacional con el objetivo de frenar la emergencia sanitaria provocada por la pandemia de enfermedad por coronavirus establecido inicialmente por un periodo de 15 días naturales mediante el Real Decreto 463/2020, disponiendo como autoridad competente el Gobierno de España y cuyo ejercicio se desarrolla a través de los Ministros de Defensa, del Interior, de Transportes, Movilidad y Agenda Urbana y de Sanidad, bajo la dirección de la Presidencia del Gobierno.



Centro de Coordinación de Alertas y Emergencias Sanitarias

Tabla 2. Casos de COVID-19 que han precisado hospitalización, ingreso en UCI y fallecidos (total y con fecha de hospitalización/ingreso en UCI/fallecimiento en los últimos 7 días) por Comunidades Autónomas en España a 19.06.2020 (datos consolidados a las 14:00 horas del 20.06.2020).

CCAA	Casos que han precisado hospitalización		Casos que han ingresado en UCI		Fallecidos	
	Total	Con fecha de ingreso en los últimos 7 días	Total	Con fecha de ingreso en UCI en los últimos 7 días	Total	Con fecha de defunción en los últimos 7 días
Andalucía	6.316	5	789	0	1.426	1
Aragón	2.682	3	273	0	911	2
Asturias	1.117	0	129	0	333	1
Baleares	1.170	0	169	0	224	0
Canarias	953	2	185	1	162	0
Cantabria	1.053	1	80	0	216	0
Castilla La Mancha	9.405	8	660	0	3.022	2
Castilla y León	8.752	17	625	2	2.776	7
Cataluña	29.310	16	2.985	2	5.666	5
Ceuta	14	0	4	0	4	0
C. Valenciana	5.806	4	742	0	1.431	2
Extremadura	1.772	0	138	0	519	0
Galicia	2.935	2	336	0	619	0
Madrid	42.324	37	3.602	1	8.416	13
Melilla	45	1	3	0	2	0
Murcia	679	0	112	0	147	0
Navarra	2.044	1	136	0	528	0
País Vasco	6.990	4	578	0	1.555	3
La Rioja	1.488	0	91	0	365	0
ESPAÑA	124.855	101	11.637	6	28.322	36

Los casos confirmados no provienen de la suma de pacientes hospitalizados, curados y fallecidos, ya que no son excluyentes. Pacientes fallecidos y curados pueden haber precisado hospitalización y por tanto computar en ambos grupos. Los pacientes que han precisado UCI también computan en los pacientes que han requerido hospitalización.

Ilustración 10. Casos de COVID-19 que han precisado hospitalización, ingreso en UCI y fallecidos por Comunidades Autónomas en España a 19/06/2020

La principal consecuencia de este mecanismo constitucional es la limitación a la libertad de circulación de los ciudadanos, quienes solo pueden circular por las vías públicas para la adquisición de alimentos, medicamentos y productos de primera necesidad, para acudir a centros sanitarios, al lugar de trabajo o entidades financieras o aseguradoras, para la asistencia a personas mayores, menores, dependientes o especialmente vulnerables, así como por causas de fuerza mayor. Igualmente, la declaración del estado de alarma supuso la suspensión de la apertura al público de los locales y establecimientos minoristas, excepto aquellos relacionados con la venta de alimentos, de productos sanitarios e higiénicos, de prensa y papelería, de tecnología, gasolineras, estancos, tintorerías y lavanderías, siempre y cuando tomaran precauciones para evitar la aglomeración de personas.

En este mismo ámbito, se suspendieron las actividades de hostelería y restauración, pero pudiendo continuar con las entregas a domicilio. Este mecanismo constitucional es la segunda vez que se utiliza en España, tras el decretado con motivo de la crisis de los controladores aéreos de 2010, si bien en aquella ocasión no se limitaron derechos a la ciudadanía general.

4.1.8 Fases del Desconfinamiento

Fase 0: Preparación de la desescalada

- Entró en vigor el 4 de mayo en la mayoría del país.

Además de las medidas de desconfinamiento del coronavirus en todo el país que ya se han aprobado como salida de niños o prácticas para hacer deporte en la calle, se añade:

- Apertura de locales con cita previa o establecimientos para la recogida a domicilio
- Comenzarán los entrenamientos profesionales e individuales

Fase 1: Fase Inicial

- Entró en vigor el 11 de mayo en la mayoría de los territorios.
- En la fase 1 del plan de desescalada del coronavirus se permitió en cada espacio territorial el inicio parcial de ciertas actividades bajo condiciones estrictas de seguridad, excepto grandes parques comerciales. Apertura de terrazas con limitación de aforo y apertura de locales.
- En la apertura de estos locales, se estableció un horario especial para mayores de 65 años.
- Apertura de hoteles y alojamientos turísticos excluyendo zonas comunes.
- Entrenamiento medio en ligas profesionales.
- En el transporte público, el uso de la mascarilla estaba altamente recomendado.
- Apertura de lugares de culto
- Apertura de Centros de Alto Rendimiento

Fase 2: Fase Intermedia

- En la fase 2 de la desescalada del coronavirus en España se abrieron locales, solo para servicio en mesas.
- El curso escolar se restablecerá en septiembre, aunque se podrán reabrir centros educativos con tres propósitos: actividades de refuerzo, garantizar que los niños menores de seis años puedan acudir a los centros si los padres tienen que acudir a trabajar y para realizar la EBAU.
- Se podrán realizar actos culturales con menos de 50 personas en lugares cerrados.
- Al aire libre se podrán reunir hasta 400 personas sentadas

Fase 3: Fase avanzada

- Se flexibilizará la movilidad general, pero se recomendará el uso de la mascarilla.
- En restauración se limitarán algo más las restricciones de aforo, pero con estricta separación entre el público.

Al concluir esta última fase del plan de desescalada del coronavirus y todas las medidas de desconfinamiento en España, que Pedro Sánchez espera que sea a finales de junio según el calendario de desescalada planteado, entraríamos ya en la nueva normalidad.



Nota.- Este cronograma es orientativo y no tiene carácter exhaustivo. Las decisiones y fechas concretas sobre el efectivo levantamiento de toda limitación establecida durante el estado de alarma se determinarán a través de los correspondientes instrumentos jurídicos.

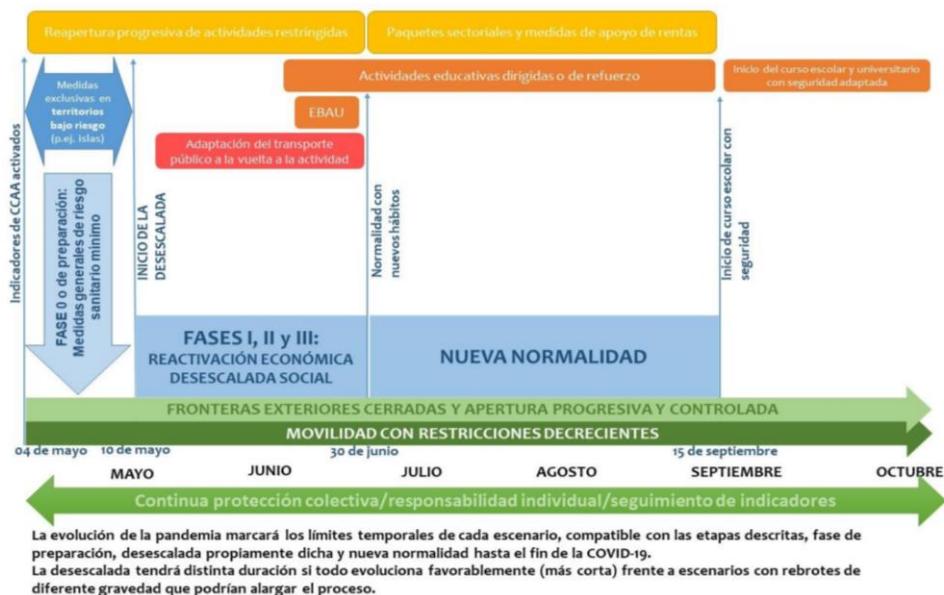


Ilustración 11. Cronograma de Confinamiento, Desescalada y Nueva Normalidad

4.2 Comprensión de los datos: Data Engineering, poniendo orden en el caos de los datos

4.2.1 Fuentes de datos

Elegimos los datos del ayuntamiento y la comunidad de Madrid tanto de calidad del aire como meteorológicos. Los datos origin vienen todos con el mismo formato representando para cada hora en cada estación de medida la magnitud a medir.

Las estaciones son los lugares dentro del municipio donde se miden las magnitudes de los contaminantes de calidad del aire o de las magnitudes meteorológicas.

Datos horarios y en tiempo real:

Cada registro está estructurado de la siguiente forma:

PROVINCIA	MUNICIPIO	ESTACION	MAGNITUD	PUNTO_MUESTREO	ANO	MES	DIA	H01	V01	H02	V02
28	79	104	82	28079104_82_98	2019	1	1	23	V	17	V

Ilustración 12. Muestra de formato de dataset original

- Las estaciones dentro del municipio de Madrid y en el resto de los municipios de la comunidad de Madrid:

ANEXO I. CÓDIGOS DE ESTACIONES

CÓDIGO	ESTACIÓN
28079102	J.M.D. Moratalaz
28079103	J.M.D. Villaverde
28079104	E.D.A.R. La China
28079106	Centro Mpal. De Acústica
28079107	J.M.D. Hortaleza
28079108	Peñagrande
28079109	J.M.D. Chamberí
28079110	J.M.D. Centro
28079111	J.M.D. Chamartín
28079112	J.M.D. Vallecas 1
28079113	J.M.D. Vallecas 2
28079114	Matadero 01
28079115	Matadero 02
28079004	Plaza España
28079008	Escuelas Aguirre
28079016	Arturo Soria
28079018	Farolillo
28079024	Casa de Campo
28079035	Plaza del Carmen
28079036	Moratalaz
28079038	Cuatro Caminos
28079039	Barrio del Pilar
28079054	Ensanche de Vallecas
28079056	Plaza Elíptica
28079058	El Pardo
28079059	Juan Carlos I

Ilustración 13. Muestra de códigos de estaciones

CÓDIGO NACIONAL	CÓDIGO MUNICIPIO	NOMBRE ESTACIÓN
28005002	5	ALCALÁ DE HENARES
28006004	6	ALCOBENDAS
28007004	7	ALCORCÓN
28009001	9	ALGETE
28013002	13	ARANJUEZ
28014002	14	ARGANDA DEL REY
28016001	16	EL ATAZAR
28045002	45	COLMENAR VIEJO
28047002	47	COLLADO VILLALBA
28049003	49	COSLADA
28058004	58	FUENLABRADA
28065014	65	GETAFE
28067001	67	GUADALIX DE LA SIERRA
28074007	74	LEGANÉS
28080003	80	MAJADAHONDA
28092005	92	MÓSTOLES
28102001	102	ORUSCO DE TAJUNA
28120001	120	PUERTO DE COTOS
28123002	123	RIVAS-VACIAMADRID
28133002	133	SAN MARTÍN DE VALDEIGLESIAS
28148004	148	TORREJÓN DE ARDOZ
28161001	161	VALDEMORO
28171001	171	VILLA DEL PRADO
28180001	180	VILLAREJO DE SALVANÉS

Ilustración 14. Muestra de Municipios incluidos en el estudio

CÓDIGO MAGNITUD	DESCRIPCIÓN MAGNITUD	CÓDIGO TÉCNICA DE MEDIDA	DESCRIPCIÓN TÉCNICA DE MEDIDA	UNIDAD	DESCRIPCIÓN UNIDAD
1	Dióxido de azufre	38	Fluorescencia ultravioleta Espectrometría infrarroja no dispersiva	µg/m³	microgramos por metro cúbico
6	Monóxido de carbono	48		mg/m³	miligramos por metro cúbico
7	Monóxido de nitrógeno	8	Quimioluminiscencia	µg/m³	microgramos por metro cúbico
8	Dióxido de nitrógeno	8	Quimioluminiscencia	µg/m³	microgramos por metro cúbico
9	Partículas en suspensión < PM2,5	49	Absorción beta	µg/m³	microgramos por metro cúbico
10	Partículas en suspensión < PM10	49	Absorción beta	µg/m³	microgramos por metro cúbico
12	Óxidos de nitrógeno	8	Quimioluminiscencia	µg/m³	microgramos por metro cúbico
14	Ozono	6	Absorción ultravioleta	µg/m³	microgramos por metro cúbico
20	Tolueno	59	Cromatografía de gases	µg/m³	microgramos por metro cúbico
22	Black Carbon	7	Absorción de luz	µg/m³	microgramos por metro cúbico
30	Benceno	59	Cromatografía de gases	µg/m³	microgramos por metro cúbico
42	Hidrocarburos totales	2	Ionización llama	mg/m³	miligramos por metro cúbico
44	Hidrocarburos no metánicos	2	Ionización llama	mg/m³	miligramos por metro cúbico
431	MetaParaXileno	59	Cromatografía de gases	µg/m³	microgramos por metro cúbico

Ilustración 15. Las magnitudes de calidad del aire y las técnicas de medida

CÓDIGO MAGNITUD	DESCRIPCIÓN MAGNITUD	CÓDIGO DE TÉCNICA DE MEDIDA	UNIDAD	DESCRIPCIÓN UNIDAD
81	Velocidad del viento	89	m/s	metros por segundo
82	Dirección del viento	89	Grd	grados
83	Temperatura	89	ºC	grados centígrados
86	Humedad relativa	89	%	porcentaje
87	Presión atmosférica	89	mbar	milibar
88	Radiación solar	89	W/m²	vatios por metro cuadrado
89	Precipitación	89	l/m²	litros por metro cuadrado

Ilustración 16. Las magnitudes meteorológicas en las estaciones

Con toda esta información comenzamos a investigar cómo se mide la calidad del aire, en el sentido de qué marca ese valor que nos comunican en distintos medios de información.

Encontramos que para medir la calidad del aire a nivel mundial, se utiliza un índice que llama ICA (Índice de Calidad del aire) cuyo valor se corresponde con el mayor valor de los contaminantes que se miden en una estación (dichos valores multiplicados por un factor, para que sean comparables). Si el municipio tiene más de una estación, el valor del ICA total será el máximo de los ICA de sus estaciones.

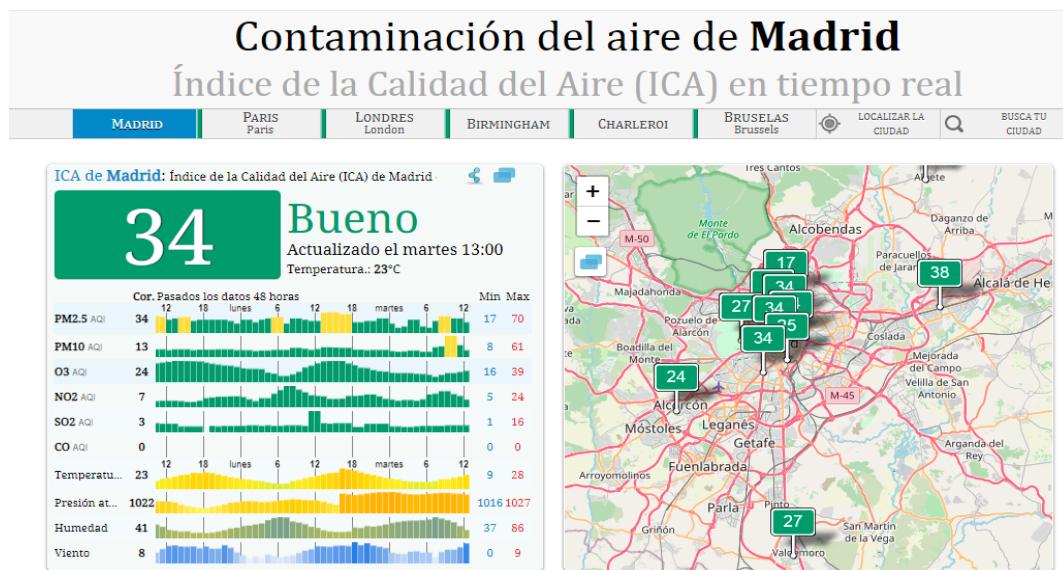


Ilustración 17. Captura de pantalla de AQICN.ORG

ICA	Calidad del Aire	Proteja su Salud
0 - 50	Buena	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este intervalo.
51 -100	Moderada	Las personas extraordinariamente sensibles deben considerar limitación de los esfuerzos físicos excesivos y prolongados al aire libre.
101-150	Dañina a la Salud de los Grupos Sensitivos	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos físicos excesivos y prolongados al aire libre.
151-200	Dañina a la Salud	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos excesivos prolongados al aire libre; las demás personas, especialmente los niños, deben limitar los esfuerzos físicos excesivos y prolongados al aire libre.
201-300	Muy Dañina a la Salud	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar todos los esfuerzos excesivos al aire libre; las demás personas, especialmente los niños, deben limitar los esfuerzos físicos excesivos al aire libre.
300+	Arriesgado	

Ilustración 18. Niveles de calidad del aire

Unas definiciones a tener en cuenta para comprender los datasets:

- **ICA:** Índice de calidad del aire.
- **VL:** Valor límite. Es un nivel fijado basándose en conocimientos científicos, con el fin de evitar, prevenir o reducir los efectos nocivos para la salud humana, para el medio ambiente en su conjunto y demás bienes de cualquier naturaleza que debe alcanzarse en un periodo determinado y no superarse una vez alcanzado.
- **Índice global:** Se corresponderá con el mayor de los índices parciales obtenidos para cada contaminante.
- **Índice parcial diario:** Es el índice obtenido para cada contaminante, asociando el valor 100 con la concentración que representa el valor límite, calculado con las concentraciones diarias u octohorarias de cada contaminante, en función de cómo esté expresado su valor límite, y con la peor de las concentraciones horarias del día en cuestión.
- **Índice parcial horario:** Es el índice obtenido para cada contaminante, asociando el valor límite con la concentración que representa el valor límite horario de cada contaminante, obtenido teniendo en cuenta la más elevada de las concentraciones horarias de cada uno de los contaminantes.

4.3. Preparación de los datos: Análisis y unificación de los datasets

Comenzamos con el análisis de los datasets y lo primero que planteamos es unir por un lado en un datasets los datos de calidad del aire y por otro los datos meteorológicos.

Creamos el campo fecha en un formato válido para el análisis de series temporales.

Para el cálculo del ICA realizamos la multiplicación de los valores de las magnitudes por cada factor:

```

#Concatenamos Datos Calidad del aire del Ayuntamiento de Madrid y Comunidad de Madrid para unificarlos.
datosCA = pd.concat([datosCACM, datosCAAM ], sort=True)

#Creamos Estacion Real
datosCA['estacion_real'] = datosCA['punto_muestreo'].str[:8]
#datosCA['estacion_real'] = datosCA.punto_muestreo.apply(lambda x: x[:8])

#Creamos Fecha
datosCA['fecha'] = datosCA.ano.apply(str) + '-' + datosCA.mes.apply(str) + '-' + datosCA.dia.apply(str)

#Obtenemos Magnitudes Calidad del Aire
urlMagnitudesCA = '/Datos/Magnitudes Calidad del Aire.csv'
magnitudesCA = pd.read_csv(urlMagnitudesCA, sep=";")

#Exportar a CSV fichero de calidad de aire
datosCA.to_csv('datosCalidadAire.csv', sep=";")

#Hacemos el merge para introducir los valores descriptivos de cada magnitud, así como los valores límite y factor de cálculo
mergeCA = datosCA.merge(magnitudesCA, left_on='magnitud', right_on='codigo_magnitud')

#mergeCA["factor_calculo_horario"] = pd.to_numeric(mergeCA["factor_calculo_horario"], downcast="float")
#mergeCA["h01"] = pd.to_numeric(mergeCA["h01"], downcast="float")

mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h01'] = mergeCA['factor_calculo_horario'] * mergeCA['h01']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h02'] = mergeCA['factor_calculo_horario'] * mergeCA['h02']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h03'] = mergeCA['factor_calculo_horario'] * mergeCA['h03']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h04'] = mergeCA['factor_calculo_horario'] * mergeCA['h04']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h05'] = mergeCA['factor_calculo_horario'] * mergeCA['h05']

```

Ilustración 19. Ejemplo de código para el cálculo de ICA parcial

Estos son los factores de cálculo de los 5 contaminantes que se utilizan para calcular el ICA:

codigo_mag nitud	descripcion_mag nitud	codigo_tecnica _de_medida	descripcion_tecnica_de _medida	unidad	descripcion_uni dad	valor_límite _diario	factor_calculo _diario	valor_límite _horario	factor_calculo _horario
1 Dióxido de azufre		38 ?g/m³	microgramos por metro cúbico	SO2	125	0.800	350	0.286	
6 Monóxido de carbono		48 mg/m³	miligramos por metro cúbico	CO	10	10	10	10	
7 Monóxido de nitrógeno		8 ?g/m³	microgramos por metro cúbico	NO					
8 Dióxido de nitrógeno		8 ?g/m³	microgramos por metro cúbico	NO2			200	0.500	
9 Partículas en suspensión < PM2,5		49 ?g/m³	microgramos por metro cúbico	PM2,5					
10 Partículas en suspensión < PM10		49 ?g/m³	microgramos por metro cúbico	PM10	50	2	150	0.667	
12 Óxidos de nitrógeno		8 ?g/m³	microgramos por metro cúbico	NOX					
14 Ozono		6 ?g/m³	microgramos por metro cúbico	O3	120	0.833	180	0.556	
20 Tolueno		59 ?g/m³	microgramos por metro cúbico	TOL					
30 Benceno		59 ?g/m³	microgramos por metro cúbico	BEN					
35 Etilbenceno		59 ?g/m³	microgramos por metro cúbico	EBE					
37 Metaxileno		59 ?g/m³	microgramos por metro cúbico	MXY					
38 Paraxileno		59 ?g/m³	microgramos por metro cúbico	PXY					
39 Ortoxileno		59 ?g/m³	microgramos por metro cúbico	OXY					
42 Hidrocarburos totales				TCH					
43 Metano		2 mg/m³	miligramos por metro cúbico	CH4					
44 Hidrocarburos no metánicos		2 mg/m³	miligramos por metro cúbico	NMHC					
22 Black Carbon		7 ?g/m³	microgramos por metro cúbico						
431 MetaParaXileno		59 ?g/m³	microgramos por metro cúbico						

Ilustración 20. Contaminantes y factores de cálculo para la obtención del ICA

4.3.1. Carga de fuentes de datos

La carga inicial le hemos realizado bajando los csv de los portales de datos abiertos de la comunidad del Madrid y del ayuntamiento de Madrid, tanto para la calidad del aire como para la información meteorológica.

Adicionalmente nos bajamos de los mismo portales los datos de las direcciones y coordenadas de las estaciones de medida. Estos csv de las estaciones de medida los unificamos en un solo csv y posteriormente con un rpa buscamos las coordenadas de latitud y longitud para poder mostrarlas con Kibana.

The screenshot shows the main page of the Madrid Open Data portal. At the top, there's a banner with the text 'Tu ciudad más cerca' and 'Gracias a nuestra plataforma de datos abiertos podrás encontrar todos los datos de Madrid que necesitas para tu proyecto'. Below the banner are navigation links for 'En portada', 'Acerca de Datos Abiertos', 'Catálogo de datos', and 'Colabora'. A search bar with the placeholder '¿Qué estás buscando?' is also present. The main content area displays the 'Calidad del aire. Datos horarios años 2001 a 2020' dataset, which includes a brief description, a list of available data types, and a note about monthly updates. There are also links to the API and transparency information.

Ilustración 21. Captura de portal del ayuntamiento de Madrid (*Calidad del aire*)

The screenshot shows the 'Datos Abiertos' section of the Madrid Community portal. It features a navigation menu with 'Catálogo', 'Transparencia', and 'Participación'. The main content area displays the '2020' dataset for 'Red de Calidad del Aire'. It includes a download link ('Descargar'), a URL ('https://datos.comunidad.madrid/catalogo/dataset/a770d92c-c513-4974-b1a7-2b15be1dd91f/resource/f525015e-5484-483...'), and a note about the file being in ZIP format. Below this is a table of additional information, including metadata like last update, creation date, and file size.

Ilustración 22. Captura de portal de la Comunidad de Madrid (*Calidad el aire*)

Ilustración 23. Captura de portal del Ayuntamiento de Madrid (Datos meteorológicos)

Datos Abiertos | Catálogo | Transparency | Participación

Home / Organizaciones / Comunidad de Madrid / Red de Calidad del Aire. ...

The screenshot shows the 'Red de Calidad del Aire. Estaciones' dataset page. On the left, there's a sidebar with sharing options (Twitter, Facebook, LinkedIn) and a license section (Creative Commons Attribution). The main content area has tabs for 'Conjunto de datos', 'grupos', and 'Flujo de Actividad'. Below these tabs, the title 'Red de Calidad del Aire. Estaciones' is displayed, followed by a brief description: 'Caracterización de las estaciones que integran la Red de Calidad del Aire de la Comunidad de Madrid'. Under the 'Datos y Recursos' section, there are two download links: 'Red de Calidad del Aire. Estaciones' in CSV format and 'Red de Calidad del Aire. Estaciones' in JSON format, each with a 'Explorar' button. At the bottom, there are three category buttons: 'Aire', 'Atmósfera', and 'Calidad del aire'.

4.3.2. Comprensión y análisis descriptivo de los datos

Una vez cargados los datos en el formato que viene, con los datos de las magnitudes en 24 columnas una para cada hora, pensamos que es mejor para estudiar series temporales desagrupar esa información y poner las magnitudes de calidad del aire en filas. De tal forma en nuestro datasets tendremos para cada día, hora, estación, magnitud de calidad del aire, su valor, su ICA, y los datos meteorológicos en ese momento.

Para esto creamos una función en Python para transformar el datasets. Convertimos el datasets en un array donde cada fila tendrá los siguientes valores, siempre y cuando la medición en esa hora de esta magnitud en esa estación sea válida (V):

1. if row.v01 == "V":
2. array.append({"id": row.fecha + "-00:00" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha + "-00:00-" + row.estacion_real, "fechahora": row.fecha + " 00:00", "fecha": row.fecha, "hora": "1", "estacion_real": row.estacion_real, "magnitud": row.magnitud, "descripcion_magnitud": row.descripcion_magnitud, "factor_calculo_horario": row.factor_calculo_horario, "ica_parcial": row.ica_h01, "valor_magnitud": row.h01, "provincia": row.provincia, "municipio": row.municipio, "dia_de_la_semana": diaDeLaSemana})

```
#Función para obtener las magnitudes de Calidad de Aire y Meteorologicas solo con valores V
def desagrupar(dataframe):
    magnitudesCalidadAire = [1,6,8,10,14]
    magnitudesMetereologicas = [81,82,83,86,87,88,89]
    array= []

    print("Empezamos con el for (esto tarda)")
    for index, row in dataframe.iterrows():
        if row.magnitud in magnitudesCalidadAire:
            #dia de la semana 0 lunes -> 6 domingo (5 y 6 fin de semana)
            diaDeLaSemana = datetime.datetime(row.ano, row.mes, row.dia).weekday()
            if row.v01 == "V":
                array.append({"id": row.fecha + "-00:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v02 == "V":
                array.append({"id": row.fecha + "-01:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v03 == "V":
                array.append({"id": row.fecha + "-02:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v04 == "V":
                array.append({"id": row.fecha + "-03:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v05 == "V":
                array.append({"id": row.fecha + "-04:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v06 == "V":
                array.append({"id": row.fecha + "-05:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
            if row.v07 == "V":
                array.append({"id": row.fecha + "-06:00-" + row.estacion_real + "-" + str(row.magnitud), "id_merge": row.fecha})
```

Ilustración 24. Ejemplo bucle para desagrupar horas

A este array con los datos de calidad lo volvemos a convertir en un dataframe y a este le añadimos las columnas de las variables meteorológicas.

```
#Creamos arrays Datos Metereológicos
array81 = desagrupar(mergeDM[mergeDM["magnitud"]== 81])
array82 = desagrupar(mergeDM[mergeDM["magnitud"]== 82])
array83 = desagrupar(mergeDM[mergeDM["magnitud"]== 83])
array86 = desagrupar(mergeDM[mergeDM["magnitud"]== 86])
array87 = desagrupar(mergeDM[mergeDM["magnitud"]== 87])
array88 = desagrupar(mergeDM[mergeDM["magnitud"]== 88])
array89 = desagrupar(mergeDM[mergeDM["magnitud"]== 89])
```

Ilustración 25. Unión con las variables meteorológicas

```
print("Creamos dataframes")
datosDefinitivos = pd.DataFrame(arrayCA)
df81 = pd.DataFrame(array81)
df82 = pd.DataFrame(array82)
df83 = pd.DataFrame(array83)
df86 = pd.DataFrame(array86)
df87 = pd.DataFrame(array87)
df88 = pd.DataFrame(array88)
df89 = pd.DataFrame(array89)
```

Ilustración 26. Creamos los dataframe.

Este datasets llamado **datosDefinitivos** es el que empezaremos a procesar.

4.3.3. Pre-procesado de datos

El pre-procesamiento y análisis de los datos lo hacemos en **Google Colab** con la funcionalidad que tiene de trabajar con notebook.

Parte 1: Carga de datos:

```
# Cargamos las librerías que vamos a usar.
import pandas as pd
import datetime
from fbprophet import Prophet
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline
from scipy.stats import norm
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns

#+ /usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
# pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
```

Ilustración 27. Carga de datos y librerías en google.colab

Importamos los datos con google.colab:

```
[ ] import pandas as pd
from google.colab import files

uploaded = files.upload()

[ ] # Carga dataframe
datos_original = pd.read_csv('datosdefinitivos.csv',sep=';')
```

Ilustración 28. Importación de datos en google.colab

Visualizamos el dataframe :

```
VISUALIZACIÓN DATAFRAME

[ ] # Consultamos el dataframe
# Magnitud 81: Velocidad_viento ; Magnitud 82: Direccion_viento ; Magnitud 83: Temperatura ;
#Magnitud 86: Humedad_relativa ; Magnitud 87: Presion_atmosferica ; Magnitud 88: Radiacion_solar ;
#Magnitud 89: Precipitacion
datos_original.head()
```

	id	id_merge	fechahora	fecha	hora	estacion_real	magnitud	descripcion_magnitud	factor_calculo_horario	ica_parcial	valor_magnitud	provincia	municipio	dia_de_la_semana	valor_magnitud_81	
0	2020-1-1-00-00-	2020-1-1-00:00-	2020-1-1	2020-1-1	1	28102001	1	Dióxido de azufre		0.286	0.286	1.0	28	102	2	1.1
1	2020-1-1-00-00-	2020-1-1-00:00-	2020-1-1	2020-1-1	1	28102001	6	Monóxido de carbono		10.000	3.000	0.3	28	102	2	1.1

Ilustración 29. Visualización de datos en google.colab

Consultamos la información del dataframe:

```
: # Consultamos la informacion del dataframe
datos_original.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243723 entries, 0 to 243722
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   id                243723 non-null   object  
 1   id_merge          243723 non-null   object  
 2   fechahora         243723 non-null   object  
 3   fecha              243723 non-null   object  
 4   hora               243723 non-null   int64  
 5   estacion_real     243723 non-null   int64  
 6   magnitud           243723 non-null   int64  
 7   descripcion_magnitud  243723 non-null   object  
 8   factor_calculo_horario  243723 non-null   float64 
 9   ica_parcial        243723 non-null   float64 
 10  valor_magnitud    243723 non-null   float64 
 11  provincia          243723 non-null   int64  
 12  municipio          243723 non-null   int64  
 13  dia_de_la_semana  243723 non-null   int64  
 14  valor_magnitud_81  243723 non-null   float64 
 15  valor_magnitud_82  243723 non-null   float64 
 16  valor_magnitud_83  243723 non-null   float64 
 17  valor_magnitud_86  243723 non-null   float64 
 18  valor_magnitud_87  243723 non-null   float64 
 19  valor_magnitud_88  243723 non-null   float64 
 20  valor_magnitud_89  243723 non-null   float64 
dtypes: float64(10), int64(6), object(5)
memory usage: 39.0+ MB
```

Ilustración 30. Dataset. Tipo de datos

Renombramos las magnitudes metereológicas para no tener que estar consultando la información de a qué corresponde cada número.

```
# Consultamos el tipo de datos de cada columna.
datos_original.dtypes
```

id	object
id_merge	object
fechahora	object
fecha	object
hora	int64
estacion_real	int64
magnitud	int64
descripcion_magnitud	object
factor_calculo_horario	float64
ica_parcial	float64
valor_magnitud	float64
provincia	int64
municipio	int64
dia_de_la_semana	int64
Velocidad_viento	float64
Direccion_viento	float64
Temperatura	float64
Humedad_relativa	float64
Presion_atmosferica	float64
Radiaccion_solar	float64
Precipitacion	float64
dtype:	object

Ilustración 31. ICA. Tipo de datos de cada columna

```
#Consultamos Valores faltantes
datos_original.isnull().sum()
```

id	0
id_merge	0
fechahora	0
fecha	0
hora	0
estacion_real	0
magnitud	0
descripcion_magnitud	0
factor_calculo_horario	0
ica_parcial	0
valor_magnitud	0
provincia	0
municipio	0
dia_de_la_semana	0
Velocidad_viento	0
Direccion_viento	0
Temperatura	0
Humedad_relativa	0
Presion_atmosferica	0
Radiaccion_solar	0
Precipitacion	0
dtype:	int64

```
#Eliminamos valores Nulos
```

```
datos_filtrados = datos_original.dropna()
```

id	0
id_merge	0
fechahora	0
fecha	0
hora	0
estacion_real	0
magnitud	0
descripcion_magnitud	0
factor_calculo_horario	0
ica_parcial	0
valor_magnitud	0
provincia	0
municipio	0
dia_de_la_semana	0
Velocidad_viento	0
Direccion_viento	0
Temperatura	0
Humedad_relativa	0
Presion_atmosferica	0
Radiaccion_solar	0
Precipitacion	0
dtype:	int64

Ilustración 32. ICA. Comprobaciones datos

Hacemos un filtrado quedándonos sólo con los que tienen un valor ICA distinto a 0

```
datos_filtrados.loc[:, (datos_filtrados == 0).all()]
datos_filtrados = datos_filtrados[datos_filtrados['ica_parcial'] != 0]
```

```
datos_filtrados.head()
```

	id	id_merge	fechahora	fecha	hora	estacion_real	magnitud	descripcion_magnitud	factor_calculo_horario	ica_parcial	...	provincia	municipio	di
0		2020-1-1-00:00:28102001-1	2020-1-1-00:00:28102001	2020-1-1	00:00	1	28102001	1	Dióxido de azufre	0.286	0.286	...	28	102
1		2020-1-1-00:00:28102001-6	2020-1-1-00:00:28102001	2020-1-1	00:00	1	28102001	6	Monóxido de carbono	10.000	3.000	...	28	102
2		2020-1-1-00:00:28102001-8	2020-1-1-00:00:28102001	2020-1-1	00:00	1	28102001	8	Dióxido de nitrógeno	0.500	4.000	...	28	102

Ilustración 33. ICA. Filtrado ICA distinto 0

Visualizamos de forma gráfica la distribución del valor del ICA (índice parcial de calidad del aire) de cada uno de los contaminantes. Cómo se observa el Ozono toma su valor máximo en torno a las 15:00 y su valor mínimo con la salida del sol 7:00. Aquí también se puede observar que los valores más altos de las 5 magnitudes corresponde a cualquier hora de día al Ozono, por tanto va a ser la magnitud que más va a afectar en el dato del ICA. Es decir que tengamos un dato de calidad de aire malo sobre en la mayoría de las ocasiones va a ser debido al Ozono.

La siguiente magnitud con dos picos máximo es el Dióxido de nitrógeno, en torno a las 8:00 y las 20:00 es cuando toma sus valores máximos.

Cuando observamos los datos de los contaminantes en cada una de las estaciones vemos que en la ciudad de Madrid el peor valor lo suele dar el dióxido de nitrógeno, sin embargo en el resto de poblaciones de la comunidad de Madrid el mayor valor lo suele dar el ozono. Así pues el valor máximo del dióxido de nitrógeno de la ciudad de Madrid puede deberse a los momentos de mayor concentración de tráfico, por las entradas y salidas de los trabajos.

```
# Suma de ICA agrupado por Hora y Magnitudes
fig, ax = plt.subplots(figsize=(15,7))
datos_filtrados.groupby(['hora','descripcion_magnitud']).sum()['ica_parcial'].unstack().plot(ax=ax)
<matplotlib.axes._subplots.AxesSubplot at 0xc869308>
```

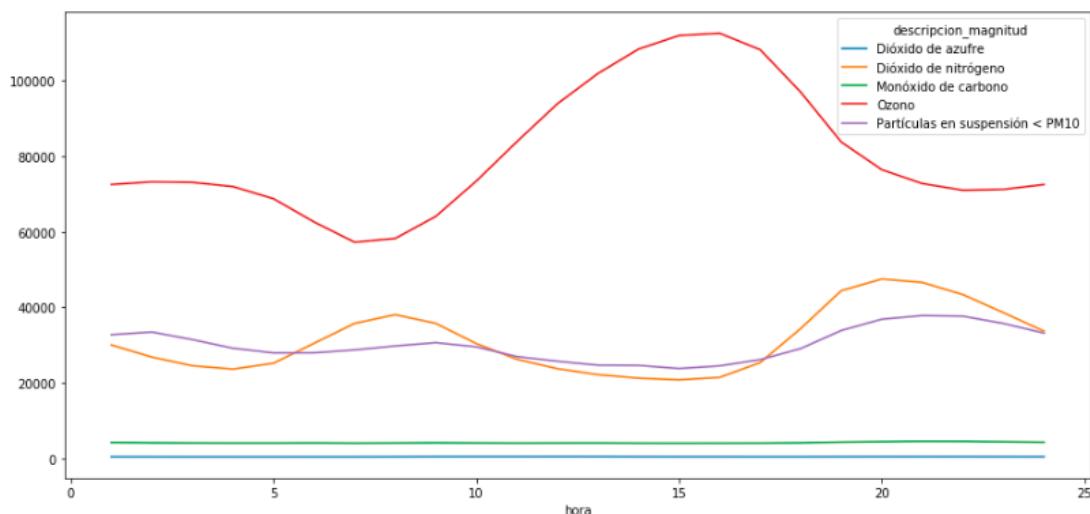


Ilustración 34. ICA. Suma de ICA Hora/magnitudes en 24 horas

Si observamos un conteo de los datos de contaminantes recogidos en estos 5 meses de estudio vemos como son los valores de Ozono y Dióxido de nitrógeno de los que tenemos más información.

Pues son los valores que se miden en casi todas las estaciones de medida.

```
print(datos_filtrados['descripcion_magnitud'].value_counts())
plt.figure(figsize=(12,5))
sns.countplot(datos_filtrados['descripcion_magnitud'])
plt.show()
```

```
Ozono          92760
Dióxido de nitrógeno 92700
Partículas en suspensión < PM10 71612
Monóxido de carbono 24904
Dióxido de azufre 24769
Name: descripción_magnitud, dtype: int64
```

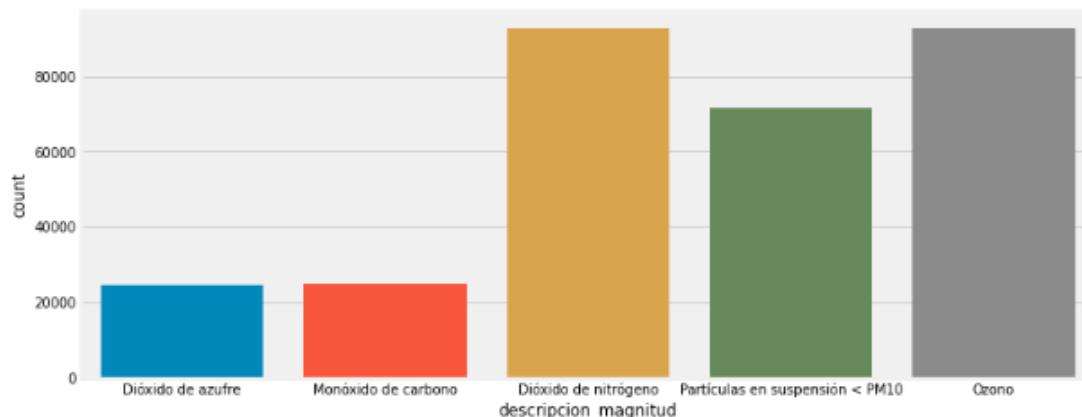


Ilustración 35. ICA. Mediciones magnitudes

La serie temporal de los contaminantes nos muestra que si bien parece que desde que comienza el periodo de confinamiento a mediados de marzo los valores de todos los contaminantes sufren una bajada en sus valores, sin embargo el Ozono sigue su tendencia habitual.

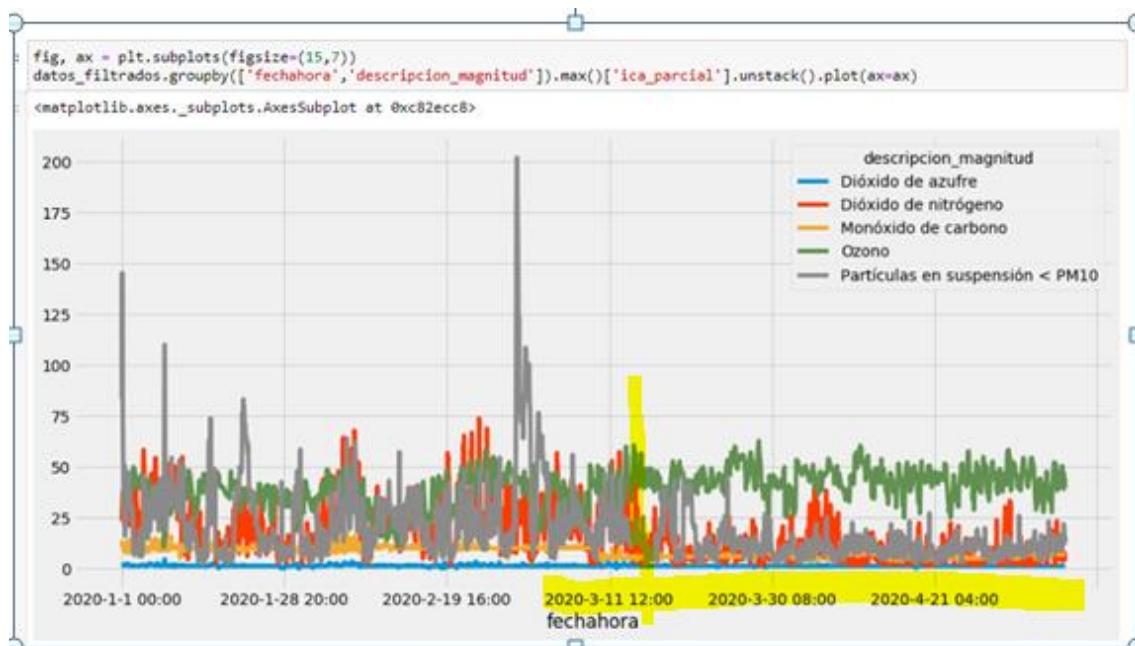


Ilustración 36. ICA. Ozono estable

Llegado a este punto queremos ver si los niveles observados, sobre todo de Ozono que es la que muestra el dato máximo de ICA parcial, son inferiores o iguales a los de los años 2018 y 2019 del mismo periodo.

En los gráficos que se muestran a continuación se aprecia como el **confinamiento** desde el 15 de marzo **ha influido significativamente en una bajada en todos los contaminantes** tanto frente a 2019 como a 2018.

Si nos fijamos en el mes de mayo 2020 vemos como los valores del Ozono comienzan a subir y se van acercando a los datos de 2019, justo cuando comienzan las fases de la desescalada y se circula más por la comunidad de Madrid.

```
datos_filtrados20182 = datos_filtrados2018[datos_filtrados2018['mes'].isin([1,2,3,4,5])]
```

```
# Suma de ICA agrupado por Hora y Magnitudes
fig, ax = plt.subplots(figsize=(15,7))
datos_filtrados20182.groupby(['año','mes','dia','descripcion_magnitud']).max()['ica_parcial'].unstack().plot(ax=ax)
<matplotlib.axes._subplots.AxesSubplot at 0x276c76c8>
```

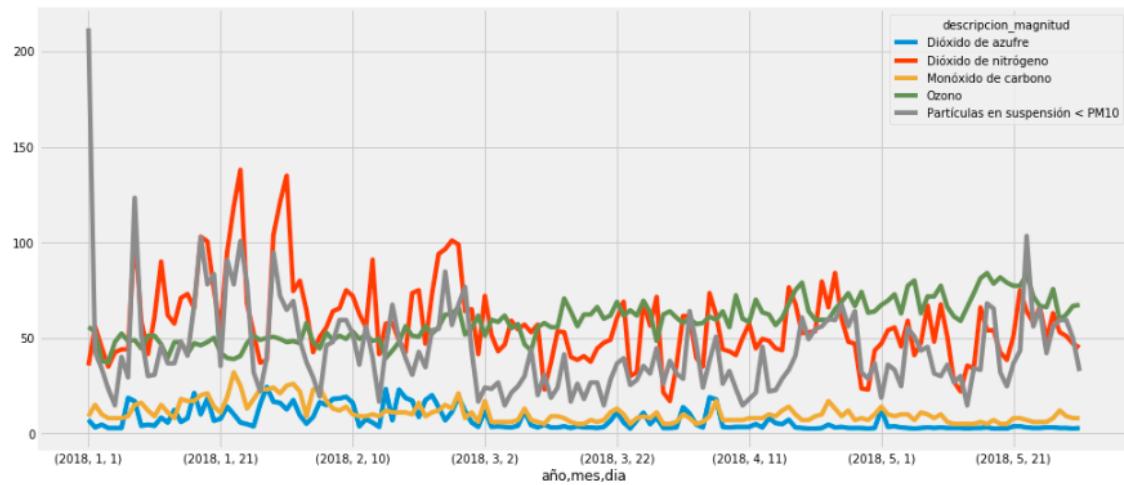


Ilustración 37. ICA. Suma de ICA Hora/magnitudes 2018

```
datos_filtrados20192 = datos_filtrados2019[datos_filtrados2019['mes'].isin([1,2,3,4,5])]
```

```
# Suma de ICA agrupado por Hora y Magnitudes
fig, ax = plt.subplots(figsize=(15,7))
datos_filtrados20192.groupby(['año','mes','dia','descripcion_magnitud']).max()['ica_parcial'].unstack().plot(ax=ax)
<matplotlib.axes._subplots.AxesSubplot at 0x13169188>
```

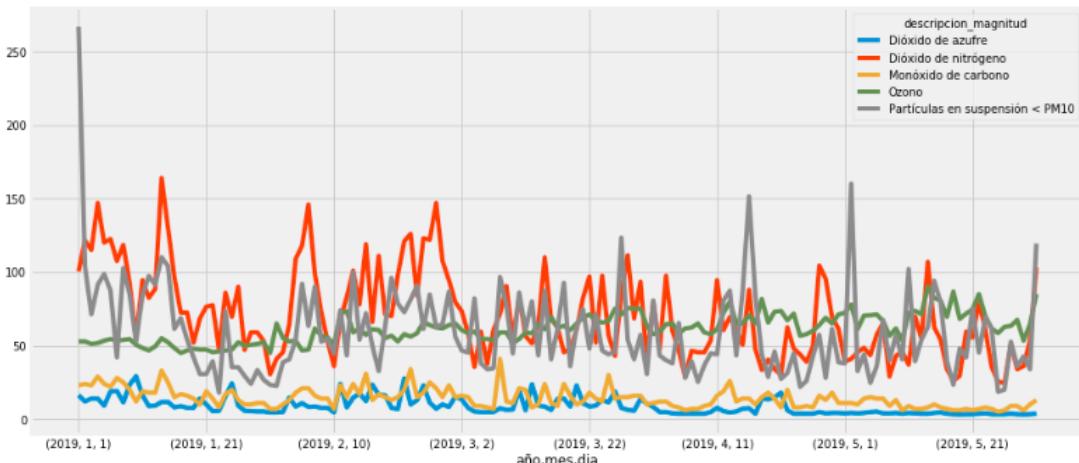


Ilustración 38. ICA. Suma de ICA Hora/magnitudes 2019

```
# Suma de ICA agrupado por Hora y Magnitudes
fig, ax = plt.subplots(figsize=(15,7))
datos_filtrados2020.groupby(['año','mes','dia','descripcion_magnitud']).max()['ica_parcial'].unstack().plot(ax=ax)
<matplotlib.axes._subplots.AxesSubplot at 0x3cc58488>
```

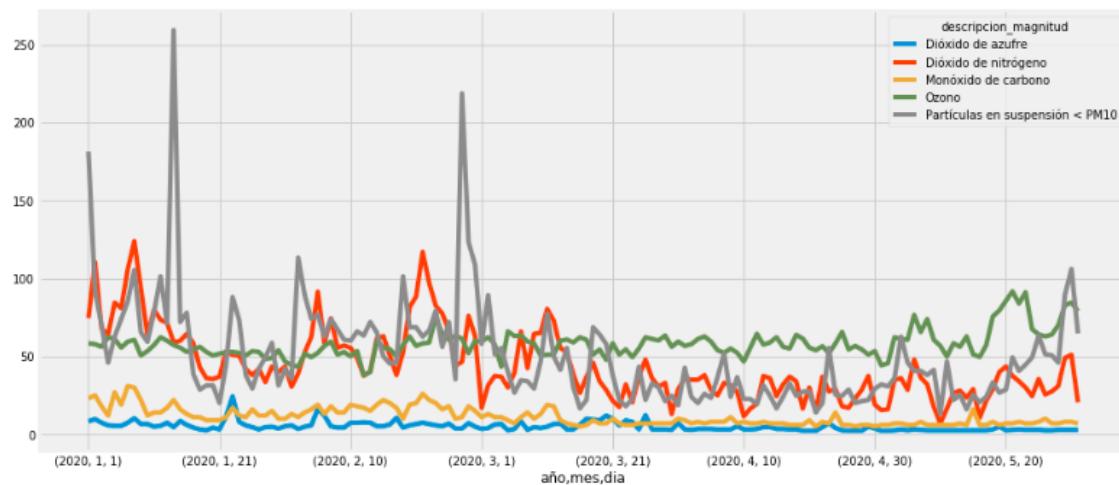


Ilustración 39. ICA. Suma de ICA Hora/magnitudes 2020

4.3.4. Análisis Descriptivo de los Datos

Visualizamos los datos que tenemos del índice de calidad del aire y de las variables meteorológicas.

4.3.4.1 Índice Calidad del Aire (ICA)

Los cálculos del Índice de Calidad del Aire se basan en el criterio establecido en el IV Seminario Nacional de Calidad del Aire de Sitges (2000).

Los compuestos que se emplean para calcular el índice de calidad son las partículas en suspensión, dióxido de azufre, dióxido de nitrógeno, monóxido de carbono y ozono. **Para cada uno de estos contaminantes se establece un índice parcial, de forma que el peor valor de los cinco definirá el índice global y, por lo tanto, la calidad del aire en el municipio de Madrid.**

ICA CALIDAD DEL AIRE Índice de Calidad del Aire($\mu\text{g}/\text{m}^3$)

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("ica_parcial", "ica_parcial", kind="scatter")
plt.show()
```

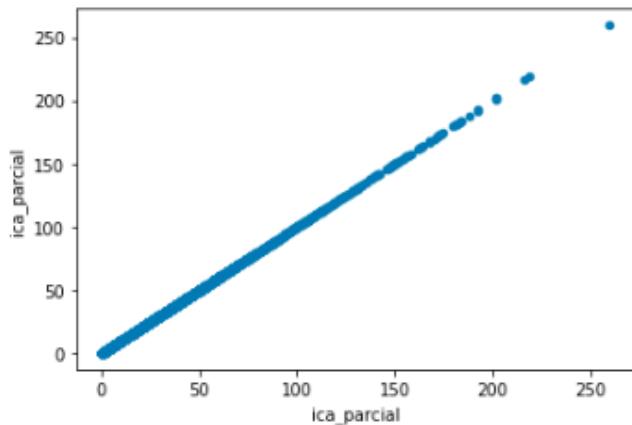


Ilustración 40. ICA. Concentración valores y outliers

```
# Consultamos los valores que más se repiten en ICA y los que menos.
ica = datos_filtrados['ica_parcial'].value_counts()
ica
```

```
0.28600    13268
4.00000    6751
1.00000    6519
2.00000    6503
5.00000    5716
...
7.26136      1
34.90568     1
1.92932      1
11.26456     1
33.98828     1
Name: ica_parcial, Length: 4398, dtype: int64
```

```
# Consultamos el promedio del valor de ICA
ica.mean()
```

```
55.41678035470669
```

```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['ica_parcial'].skew())
print("Curtosis: %f" % datos_filtrados['ica_parcial'].kurt())
```

```
Asimetría: 1.458302
Curtosis: 4.056246
```

Ilustración 41. ICA. Asimetría y curtosis

```
df = datos_filtrados.loc[:, ["fecha","ica_parcial"]]
df['fecha'] = pd.DatetimeIndex(df['fecha'])
df.dtypes

fecha          datetime64[ns]
ica_parcial    float64
dtype: object

plt.figure(figsize=(12,5))
plt.title("ICA")
ax = sns.distplot(datos_filtrados["ica_parcial"], color = 'y')
```

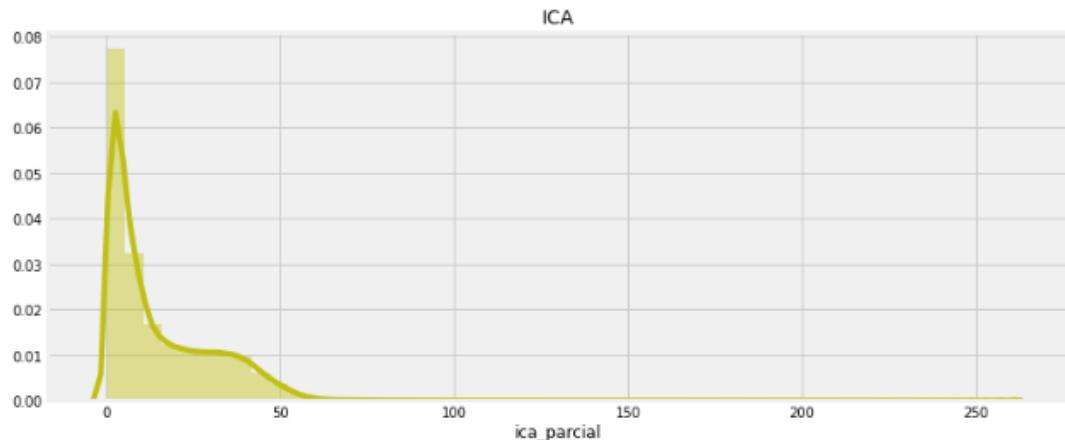


Ilustración 42. ICA. Distribución valores ICA

```
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['ica_parcial']).ica_parcial)
```

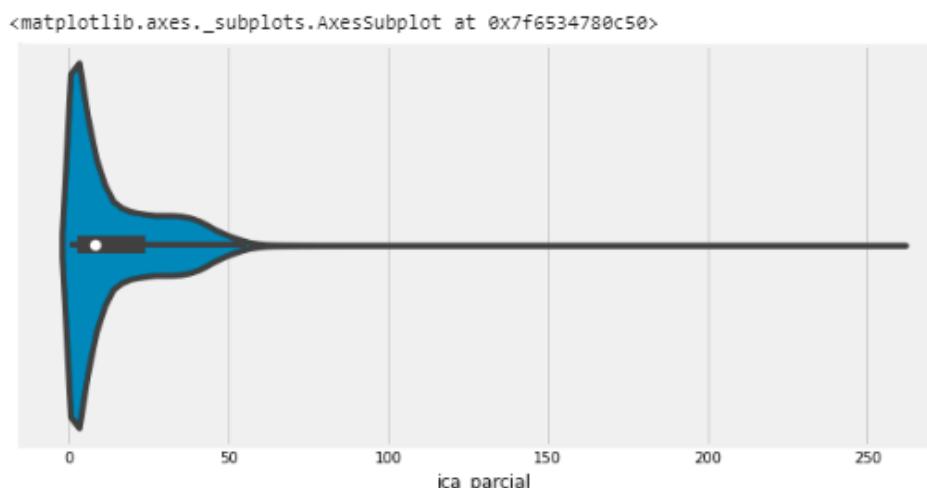


Ilustración 43. ICA. Grafica violín

Observando la media mensual del valor del índice ICA , se parecía una ligera bajada en el mes de abril, que es cuando se produjo el mayor confinamiento total de la población en la comunidad de Madrid.

```
# Mostramos los meses del análisis que son Enero, Febrero, Marzo, Abril y Mayo
df['mes'].unique()

array([1, 2, 3, 4, 5])

# Media mensual ICA
df.groupby(['mes']).mean()[['ica_parcial']]

ica_parcial
```

mes	ica_parcial
1	14.673963
2	14.944857
3	14.709132
4	13.516675
5	15.341146

Ilustración 44. Media mensual ICA

```
# Graficamos promedio ica por meses del 1(enero) al 5(mayo)
dategroup=df.groupby('mes').mean()
plt.figure(figsize=(12,5))
dategroup[['ica_parcial']].plot(x=df.fecha)
plt.title('ICA PROMEDIO MES')

Text(0.5, 1.0, 'ICA PROMEDIO MES')
```

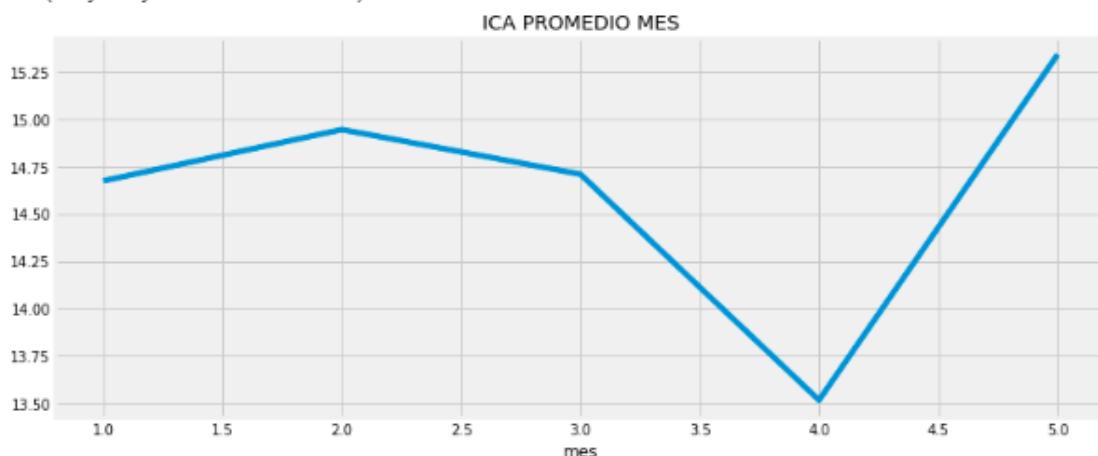


Ilustración 45. Promedio ICA mes

La distribución horaria del ICA muestra el menor dato con la salida del sol , en torno a las 7:00. y un mayor dato en las horas centrales del día en torno a las 18:00.

```
: # Valor ICA por HORAS  
dategroup=datos_filtrados.groupby('hora').mean()  
plt.figure(figsize=(12,5))  
dategroup['ica_parcial'].plot(x=datos_filtrados.fecha)  
plt.title('ICA')  
  
: Text(0.5, 1.0, 'ICA')
```

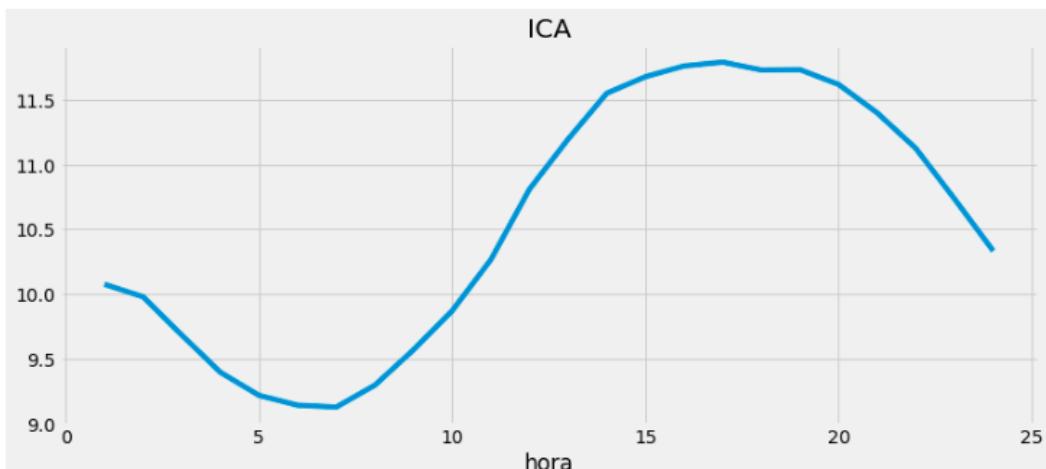


Ilustración 46. Distribución ICA horaria

4.3.4.2 Velocidad del viento

El anemómetro o anemógrafo es un aparato meteorológico utilizado para medir la velocidad del viento y así ayudar en la predicción del tiempo.

Los datos se muestran en m/s.

A continuación, mostramos los valores de la velocidad del viento en los periodos analizados:

```
#Mostramos histograma con valores velocidad de viento
plt.figure(figsize=(12,5))
plt.title("Velocidad del viento")
ax = sns.distplot(datos_filtrados["Velocidad_viento"], color = 'y')
```

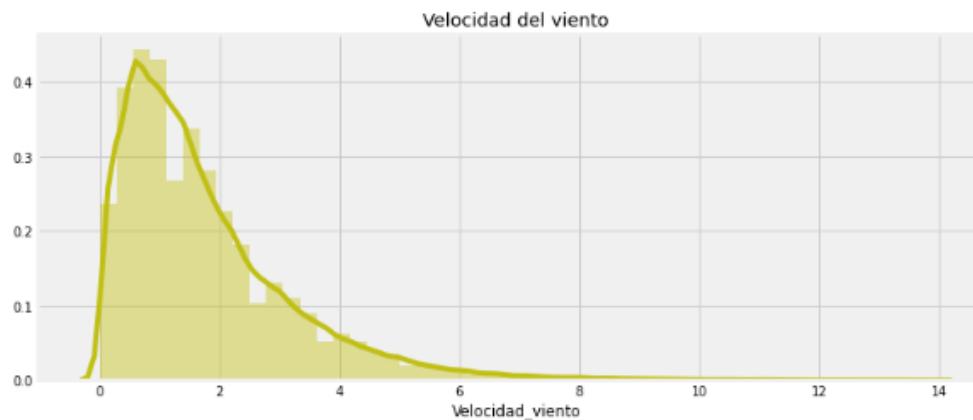


Ilustración 47. Velocidad del viento. Histograma

```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['Velocidad_viento'].skew())
print("Curtosis: %f" % datos_filtrados['Velocidad_viento'].kurt())
```

```
Asimetría: 1.793058
Curtosis: 5.008540
```

```
# Mostramos gráfico tipo violín con datos Velocidad_viento
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['Velocidad_viento']).Velocidad_viento)
```

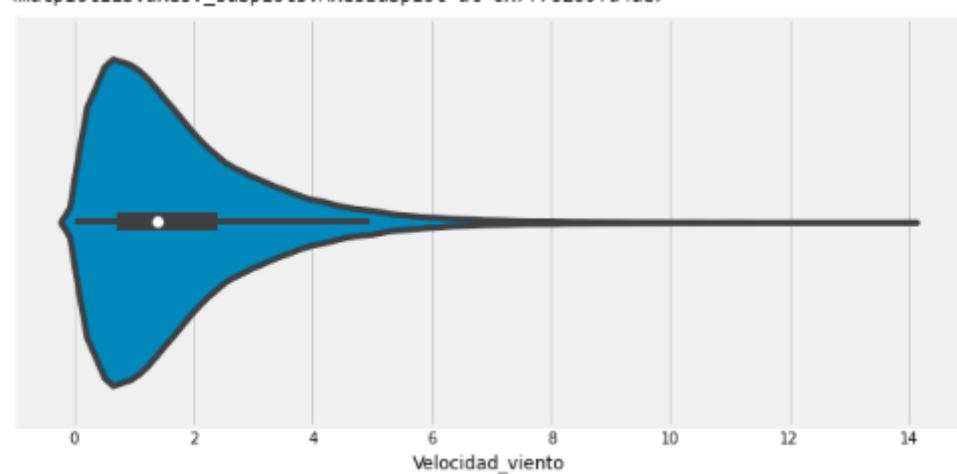


Ilustración 48. Velocidad del viento. Asimetría y curtosis

```
# Consultamos la concentración de valores y outliers  
my_plot = datos_filtrados.plot("Velocidad_viento", "Velocidad_viento", kind="scatter")  
plt.show()
```

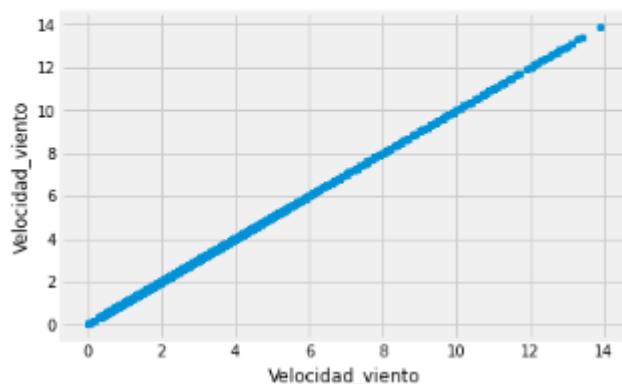
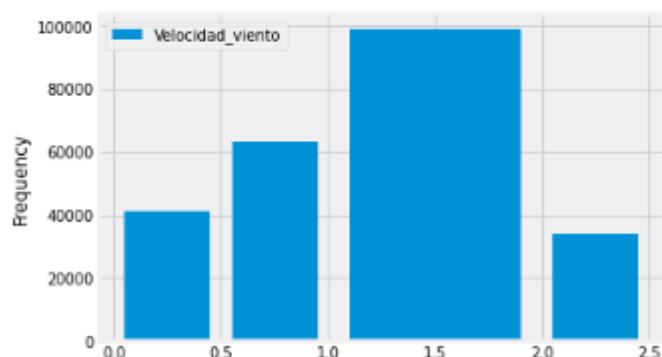


Ilustración 49. Velocidad del viento. Concentración y outliers

```
# Distribución de los valores de la magnitud Velocidad del viento  
datos_filtrados[['Velocidad_viento']].plot(kind='hist', bins=[0,0.5,1.0,2,2.0,2.5], rwidth=0.8)  
plt.show()
```



```
# Creamos dataframe con viento y Convertimos fecha en datetime  
viento = datos_filtrados.loc[:, ["fecha", "Velocidad_viento"]]  
viento['fecha'] = pd.DatetimeIndex(viento['fecha'])  
viento.dtypes
```

```
fecha          datetime64[ns]  
Velocidad_viento      float64  
dtype: object
```

```
# Consultamos valores del la magnitud viento  
viento.describe()
```

Ilustración 50. Velocidad del viento. Distribución magnitud

```
| # Consultamos valores de la magnitud viento  
viento.describe()
```

Velocidad_viento	
count	306745.000000
mean	1.765884
std	1.459266
min	0.000000
25%	0.700000
50%	1.400000
75%	2.400000
max	13.900000

Ilustración 51. Velocidad del viento. Valores magnitud

4.3.4.3 Dirección del viento

El viento es la variable de estado de movimiento del aire. En meteorología se estudia el viento como aire en movimiento tanto horizontal como verticalmente. Los movimientos verticales del aire caracterizan los fenómenos atmosféricos locales, como la formación de nubes de tormenta.

Los movimientos horizontales son los que más importancia meteorológica y trascendencia práctica tienen para la navegación. Este movimiento horizontal del aire es el que se conoce como "viento".

Los datos que se muestran están en grados centígrados de 0 a 359°.

```
#Mostramos histograma con valores dirección del viento
plt.figure(figsize=(12,5))
plt.title("Dirección del viento")
ax = sns.distplot(datos_filtrados["Direccion_viento"], color = 'y')
```

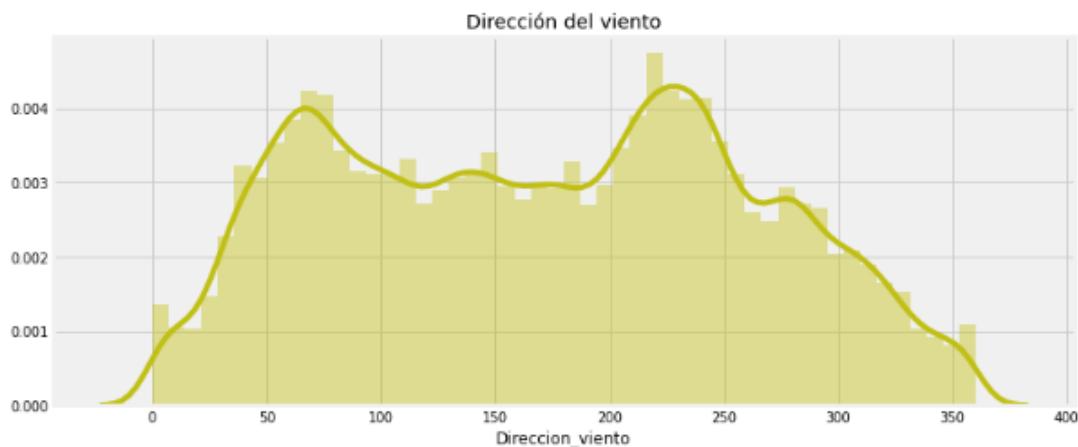


Ilustración 52. Dirección del viento. Histograma

```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['Direccion_viento'].skew())
print("Curtosis: %f" % datos_filtrados['Direccion_viento'].kurt())
```

Asimetría: 0.053952
Curtosis: -1.062406

```
# Mostramos gráfico tipo violín con datos Direccion_viento
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['Direccion_viento']).Direccion_viento)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f76201180b8>

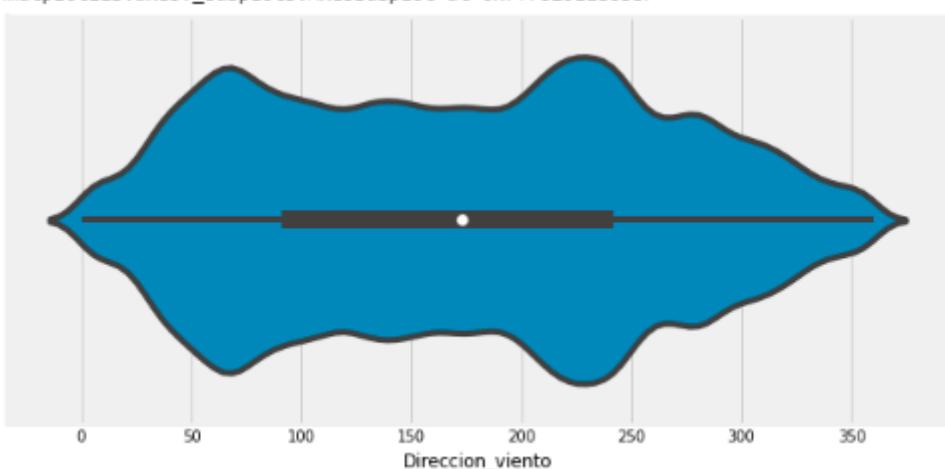


Ilustración 53. Dirección del viento. Asimetría y curtosis

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("Direccion_viento", "Direccion_viento", kind="scatter")
plt.show()
```

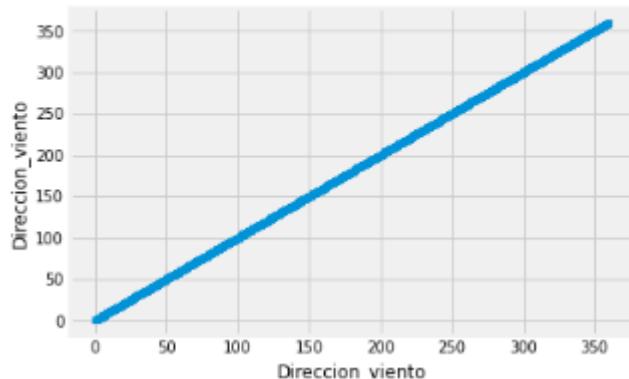


Ilustración 54. Dirección del viento. Concentración y outliers

```
# Creamos dataframe con Direccion_viento y Convertimos fecha en datetime
direccion = datos_filtrados.loc[:, ["fecha", "Direccion_viento"]]
direccion['fecha'] = pd.DatetimeIndex(direccion['fecha'])
direccion.dtypes
```

```
fecha           datetime64[ns]
Direccion_viento      float64
dtype: object
```

```
# Consultamos valores del la magnitud viento
direccion.describe()
```

Direccion_viento	
count	306745.000000
mean	171.096181
std	90.260191
min	0.000000
25%	91.000000
50%	173.000000
75%	242.000000
max	360.000000

Ilustración 55. Dirección del viento. Conversión fecha dataframe

4.3.4.4 Temperatura

Se llama **temperatura atmosférica** a uno de los elementos constitutivos del clima que se refiere al grado de calor específico del aire en un lugar y momento determinados, así como la evolución temporal y espacial de dicho elemento en las distintas zonas climáticas.

Los datos se muestran en grados centígrados (°C).

```
#Mostramos histograma con valores Temperatura  
plt.figure(figsize=(12,5))  
plt.title("Temperatura")  
ax = sns.distplot(datos_filtrados["Temperatura"], color = 'y')
```

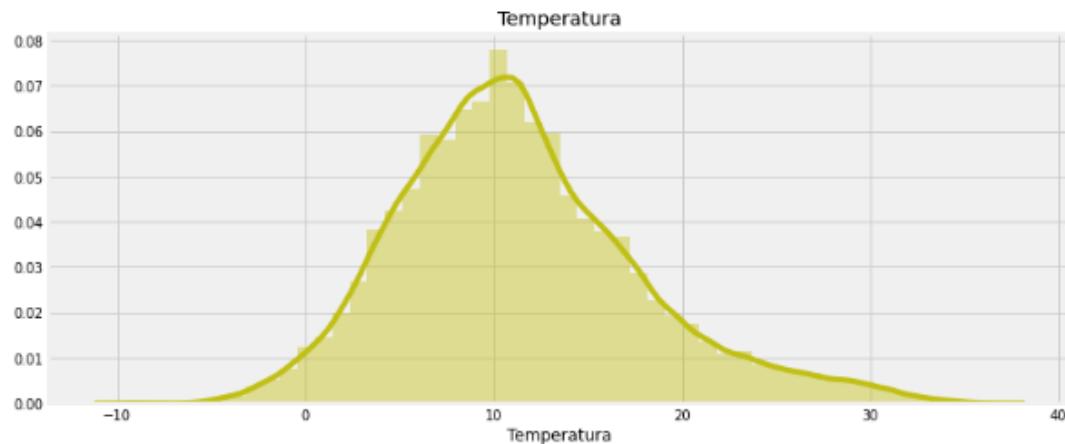


Ilustración 56. Temperatura. Histograma

```
# Asimetría y curtosis:  
print("Asimetría: %f" % datos_filtrados['Temperatura'].skew())  
print("Curtosis: %f" % datos_filtrados['Temperatura'].kurt())
```

```
Asimetría: 0.601611  
Curtosis: 0.500122
```

```
# Mostramos gráfico tipo violín con datos Temperatura  
fig, ax = plt.subplots()  
fig.set_size_inches(10,5)  
sns.violinplot(datos_filtrados.dropna(subset = ['Temperatura']).Temperatura)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f761fec7710>
```

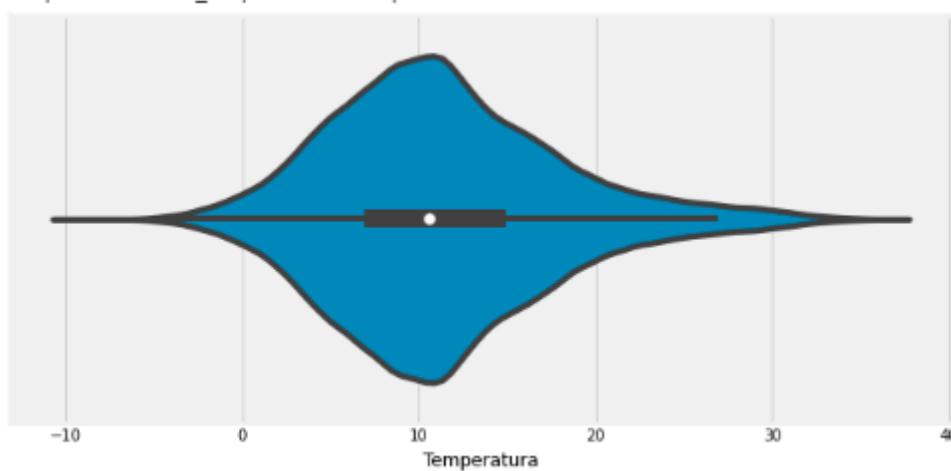


Ilustración 57. Temperatura. Gráfico violín

```
| w# Consultamos la concentración de valores y outliers  
my_plot = datos_filtrados.plot("Temperatura", "Temperatura", kind="scatter")  
plt.show()
```

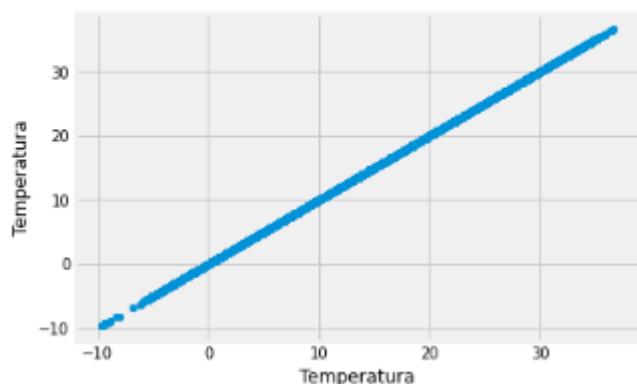


Ilustración 58. Temperatura. Concentración y outliers

4.3.4.5 Humedad relativa

El grado o cantidad de humedad de aire se mide con el higrómetro. Cuando el higrómetro marca el 100 % se dice que el aire está saturado, es decir, contiene el máximo de humedad que puede tener a la temperatura actual.

Los datos se muestran en porcentaje (%).

```
#Mostramos histograma con valores Humedad relativa  
plt.figure(figsize=(12,5))  
plt.title("Humedad_relativa")  
ax = sns.distplot(datos_filtrados["Humedad_relativa"], color = 'y')
```

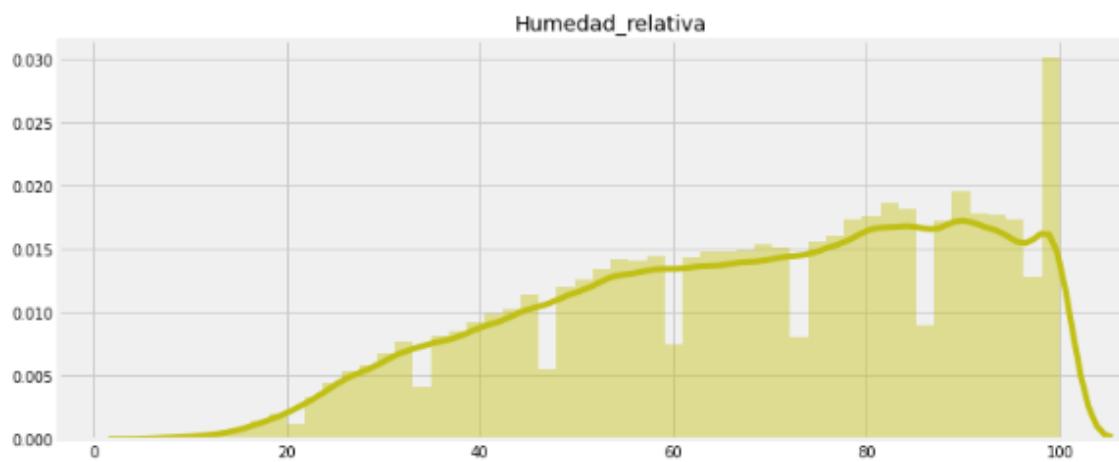


Ilustración 59. Humedad relativa. Histograma

Asimetría: -0.367189
Curtosis: -0.840936

```
# Mostramos gráfico tipo violín con datos Humedad relativa
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['Humedad_relativa']).Humedad_relativa)

<matplotlib.axes._subplots.AxesSubplot at 0x7f761fce69b0>
```

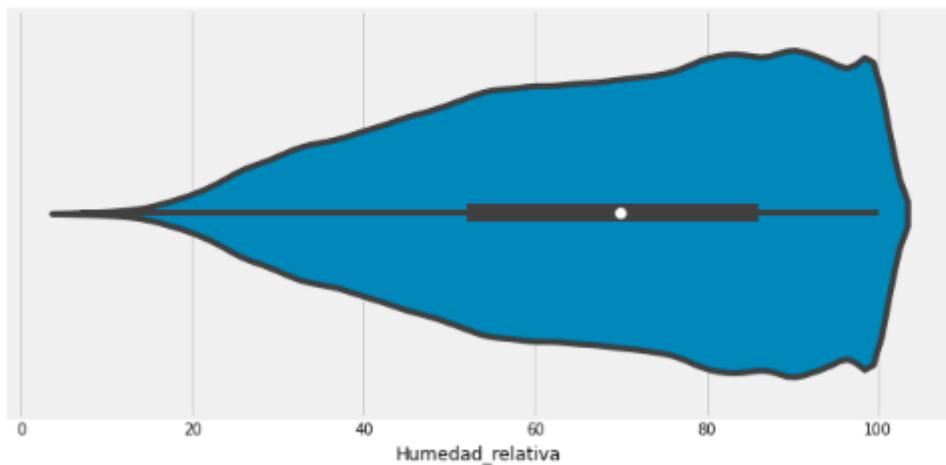
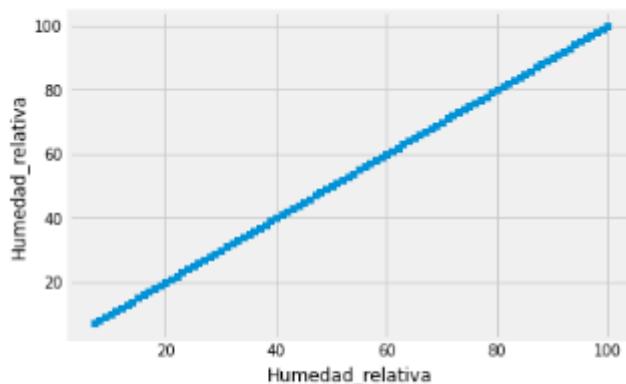


Ilustración 60. Humedad relativa. Gráfico violín

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("Humedad_relativa", "Humedad_relativa", kind="scatter")
plt.show()
```



```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['Humedad_relativa'].skew())
print("Curtosis: %f" % datos_filtrados['Humedad_relativa'].kurt())
```

Asimetría: -0.367189
Curtosis: -0.840936

Ilustración 61. Humedad relativa. Concentración y outliers

4.3.4.6 Presión atmosférica

La presión atmosférica se mide con un aparato llamado barómetro.
Los datos se muestran en mb (amilbar)

```
#Mostramos histograma con valores Presion atmosferica
plt.figure(figsize=(12,5))
plt.title("Presion_atmosferica")
ax = sns.distplot(datos_filtrados["Presion_atmosferica"], color = 'y')
```

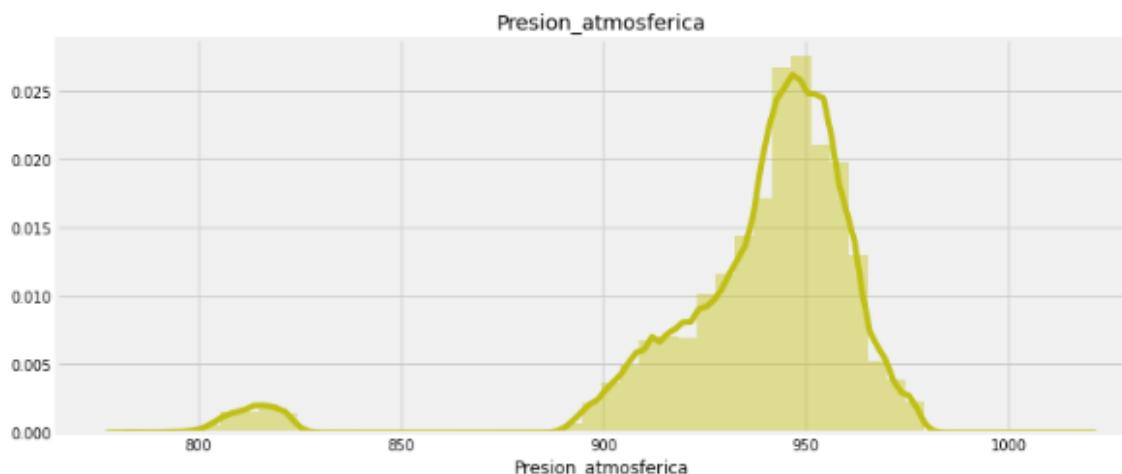


Ilustración 62. Presión atmosférica. Histograma

```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['Presion_atmosferica'].skew())
print("Curtosis: %f" % datos_filtrados['Presion_atmosferica'].kurt())
```

```
Asimetría: -2.651511
Curtosis: 8.872800
```

```
# Mostramos gráfico tipo violín con datos Presion atmosferica
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['Presion_atmosferica']).Presion_atmosferica)
```

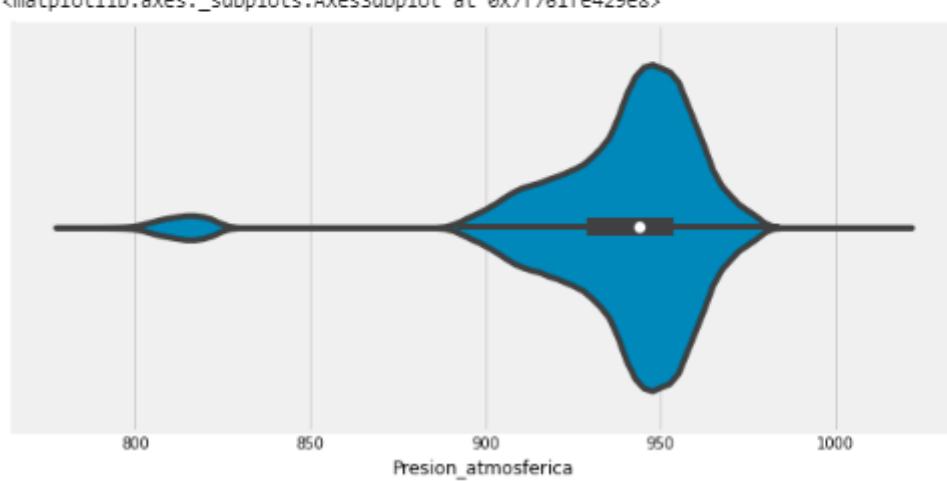


Ilustración 63. Presión atmosférica. Asimetría y curtosis

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("Presion_atmosferica", "Presion_atmosferica", kind="scatter")
plt.show()
```

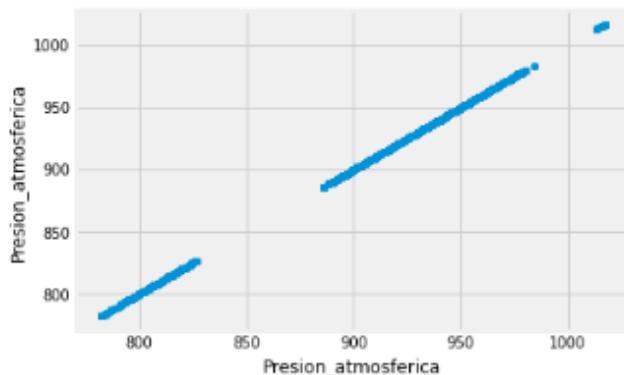


Ilustración 64. Presión atmosférica. Concentración y outliers

4.3.4.7 Radiación Solar

Se mide en superficie horizontal, mediante el sensor de radiación o piranómetro, que se sitúa orientado al sur y en un lugar libre de sombras.

La unidad de medida es vatios por metro cuadrado (W/m²).

```
#Mostramos histograma con valores Radiación Solar
plt.figure(figsize=(12,5))
plt.title("Radiacion_solar")
ax = sns.distplot(datos_filtrados["Radiacion_solar"], color = 'y')
```

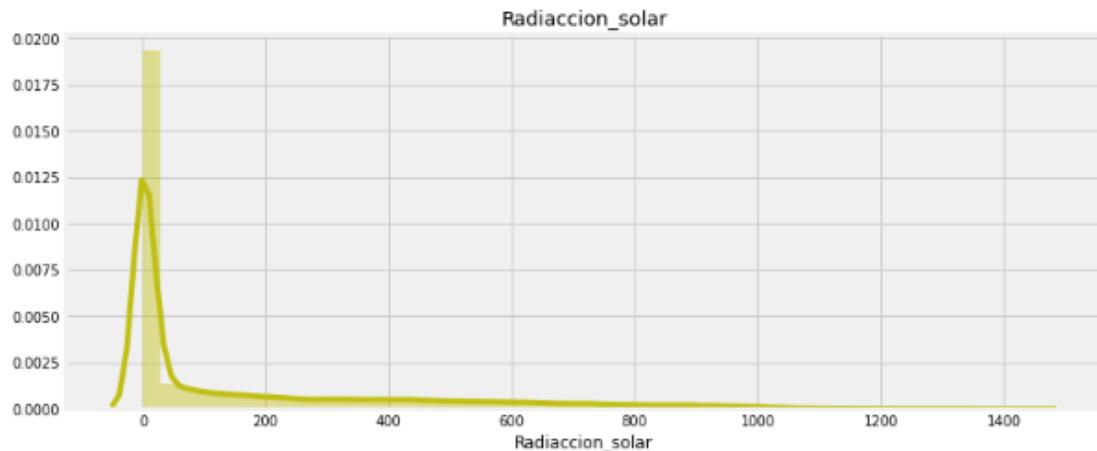


Ilustración 65. Radiación solar. Histograma

```
# Asimetría y curtosis:
print("Asimetría: %f" % datos_filtrados['Radiacion_solar'].skew())
print("Curtosis: %f" % datos_filtrados['Radiacion_solar'].kurt())
```

Asimetría: 1.607255
 Curtosis: 1.641561

```
# Mostramos gráfico tipo violín con datos Radiacion solar
fig, ax = plt.subplots()
fig.set_size_inches(10,5)
sns.violinplot(datos_filtrados.dropna(subset = ['Radiacion_solar']).Radiacion_solar)

<matplotlib.axes._subplots.AxesSubplot at 0x7f762006e7b8>
```

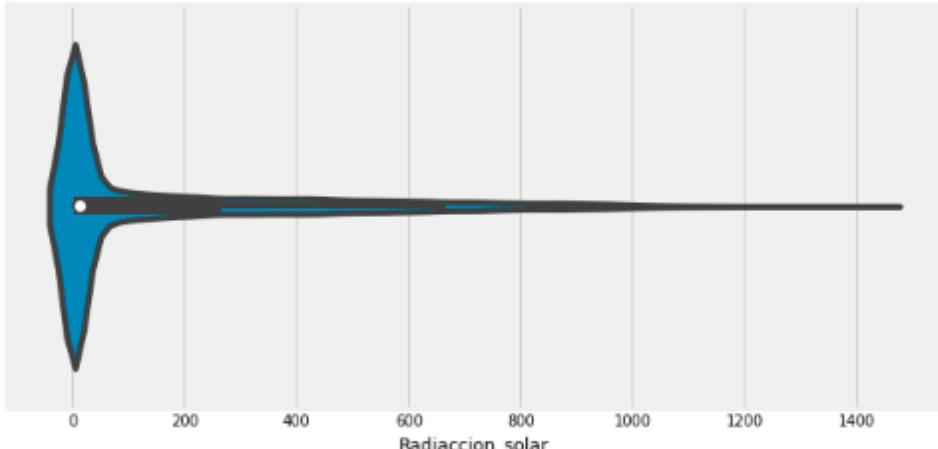


Ilustración 66. Radiación solar. Asimetría y curtosis

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("Radiacion_solar", "Radiacion_solar", kind="scatter")
plt.show()
```

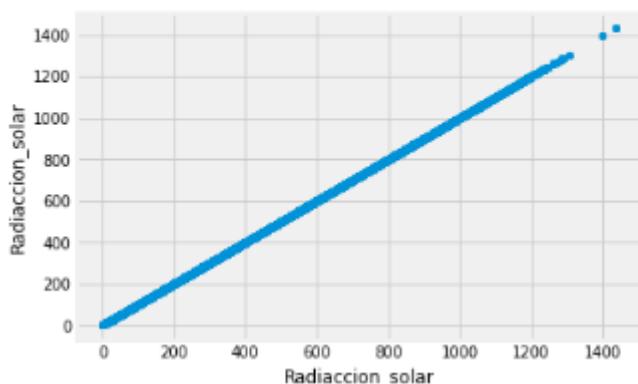


Ilustración 67. Radiación solar. Concentración y outliers

4.3.4.8 Precipitación

La precipitación se mide en **l/m²** (litros por metro cuadrado), **1 mm = 1 l/m²**. Equivale a los litros de agua de lluvia caídos en una superficie cuadrada de una longitud de un metro por cada lado con paredes verticales.

```
#Mostramos histograma con valores Precipitacion  
plt.figure(figsize=(12,5))  
plt.title("Precipitacion")  
ax = sns.distplot(datos_filtrados["Precipitacion"], color = 'y')
```

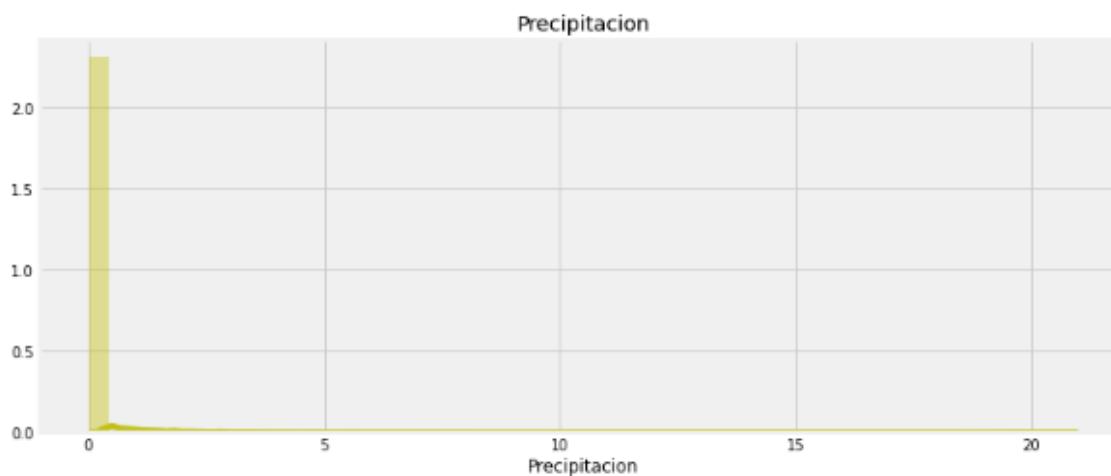


Ilustración 68. Precipitación. Histograma

```
# Asimetría y curtosis:  
print("Asimetría: %f" % datos_filtrados['Precipitacion'].skew())  
print("Curtosis: %f" % datos_filtrados['Precipitacion'].kurt())
```

```
Asimetría: 15.438756  
Curtosis: 365.609030
```

```
# Mostramos gráfico tipo violín con datos Precipitacion  
fig, ax = plt.subplots()  
fig.set_size_inches(10,5)  
sns.violinplot(datos_filtrados.dropna(subset = ['Precipitacion']).Precipitacion)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f76202946a0>
```

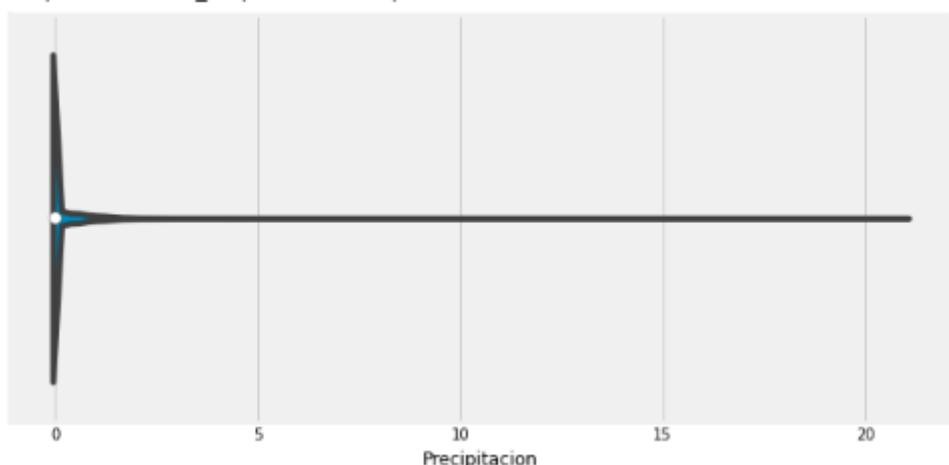


Ilustración 69. Precipitación. Asimetría y curtosis

```
# Consultamos la concentración de valores y outliers
my_plot = datos_filtrados.plot("Precipitacion", "Precipitacion", kind="scatter")
plt.show()
```

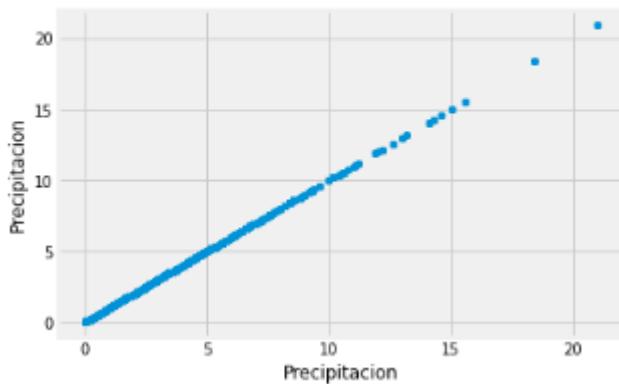


Ilustración 70. Precipitación. Concentración de valores y outliers

4.4 Modelado

4.4.1. Preparación del modelo

Para preparar los datos para el modelo comenzamos con una visualización de los datos para lo cual normalizando los datos:

```
[ ] # Normalizamos datos de ICA y magnitudes
from sklearn import preprocessing
values = ['ica_parcial', 'Velocidad_viento', 'Direccion_viento', 'Temperatura', 'Humedad_relativa','Presion_atmosferica','Radiacion_solar','Precipitacion']
x = datos_filtrados[values] #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
datos_normalizados = pd.DataFrame(x_scaled)
datos_normalizados.columns = ['ica_parcial', 'Velocidad_viento', 'Direccion_viento', 'Temperatura', 'Humedad_relativa','Presion_atmosferica','Radiacion_solar','Precipitacion']
datos_normalizados.insert(0, "Fecha", datos_filtrados["Fecha"])
datos_normalizados.head()
```

	Fecha	ica_parcial	Velocidad_viento	Direccion_viento	Temperatura	Humedad_relativa	Presion_atmosferica	Radiacion_solar	Precipitacion
0	2020-1-1	0.000000	0.079137	0.041667	0.327177	0.849462	0.828283	0.009346	0.0
1	2020-1-1	0.010472	0.079137	0.041667	0.327177	0.849462	0.828283	0.009346	0.0
2	2020-1-1	0.014330	0.079137	0.041667	0.327177	0.849462	0.828283	0.009346	0.0
3	2020-1-1	0.019485	0.079137	0.041667	0.327177	0.849462	0.828283	0.009346	0.0
4	2020-1-1	0.082561	0.079137	0.041667	0.327177	0.849462	0.828283	0.009346	0.0

Ilustración 71. Preparación de modelado

```
#datos_normalizados.describe()
datos_normalizados.dtypes

Fecha          object
ica_parcial    float64
Velocidad_viento float64
Direccion_viento float64
Temperatura    float64
Humedad_relativa float64
Presion_atmosferica float64
Radiacion_solar float64
Precipitacion   float64
dtype: object

# Correlacion entre variables
datos_normalizados.corr()

  ica_parcial  Velocidad_viento  Direccion_viento  Temperatura  Humedad_relativa  Presion_atmosferica  Radiacion_solar  Precipitacion
  ica_parcial      1.000000     -0.005117      0.029837     0.066666     -0.125989     -0.017760      0.043789     -0.003836
  Velocidad_viento -0.005117      1.000000      0.077269     0.067277     -0.179103     -0.172627      0.176801      0.029434
  Direccion_viento  0.029837      0.077269      1.000000     0.156068     -0.051785     -0.020500      0.034052     -0.020661
  Temperatura       0.066666      0.067277     0.156068      1.000000     -0.544084     0.210312      0.486563     -0.053249
  Humedad_relativa -0.125989     -0.179103     -0.051785     0.100000      1.000000     -0.045112     -0.476670      0.137374
  Presion_atmosferica -0.017760     -0.172627     -0.020500     0.210312     -0.045112      1.000000     -0.028128     -0.109487
  Radiacion_solar    0.043789      0.176901      0.034052     0.486563     -0.476670     -0.028128      1.000000     -0.057845
  Precipitacion       -0.003836     0.029434     -0.020661     -0.053249      0.137374     -0.109487     -0.057845      1.000000
```

Se observan muchos datos atípicos en todas las variables.

```
### Consultamos diagramas box y whisker (cajas y bigotes)
boxplot = datos_normalizados.boxplot(grid=False, rot=45, fontsize=12)
```

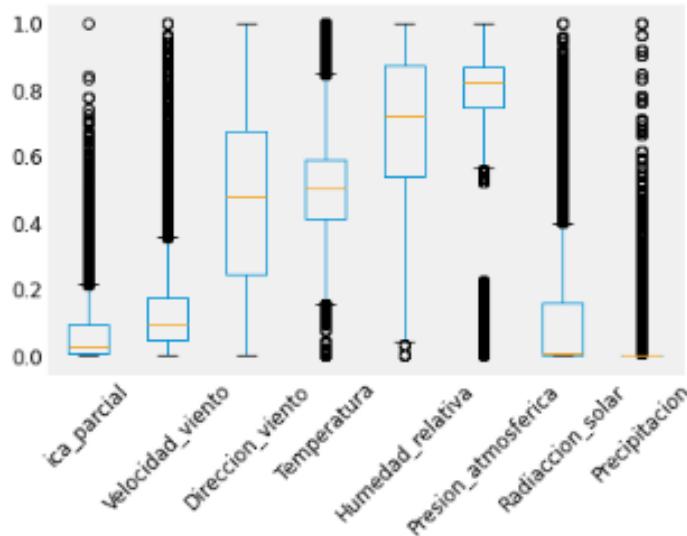


Ilustración 72. Diagrama box y whisker

En la matriz de correlaciones no se observa ninguna correlación fuerte entre el índice ICA y las variables metereológicas.

```
# Matriz de correlación de las magnitudes
corrmat = datos_normalizados.corr(method='pearson')
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
```

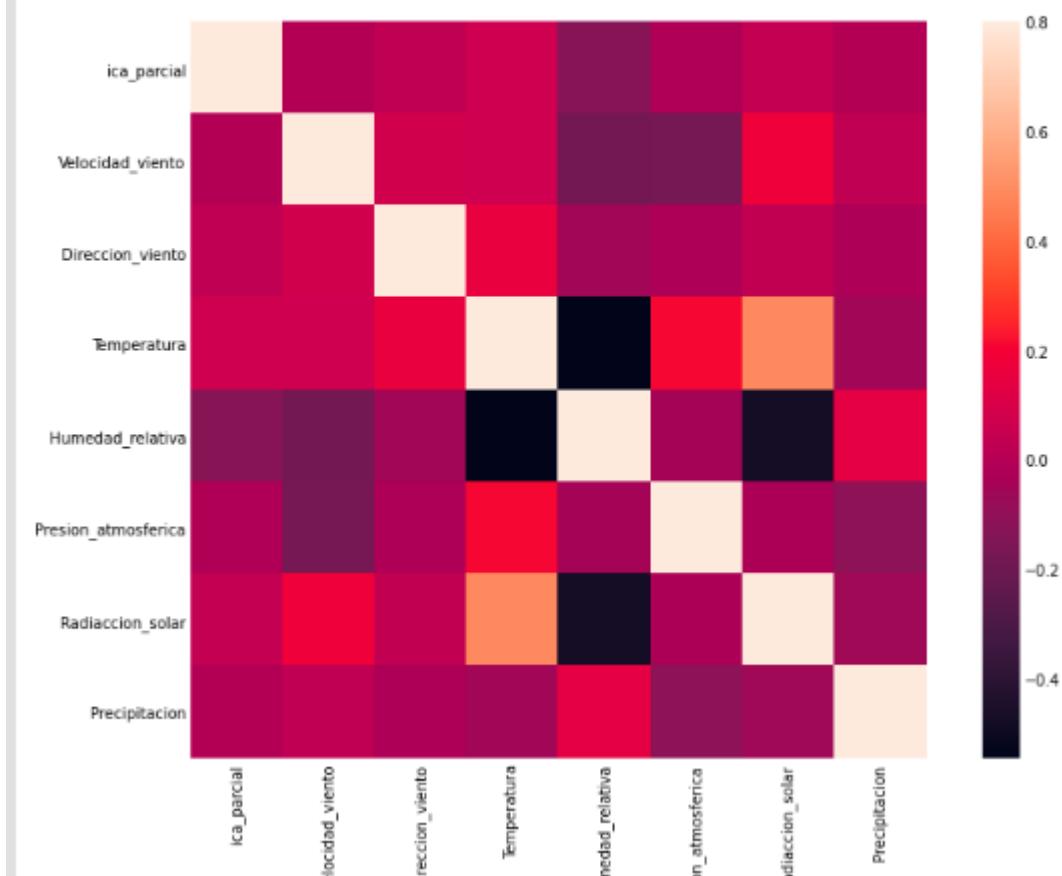


Ilustración 73. Matriz de correlación

```
#Correlacion con datos
f,ax = plt.subplots(figsize=(9, 9))
sns.heatmap(datos_normalizados.corr(), annot=True, linewidths=.5, fmt= '.1f', ax=ax)
plt.show()
```

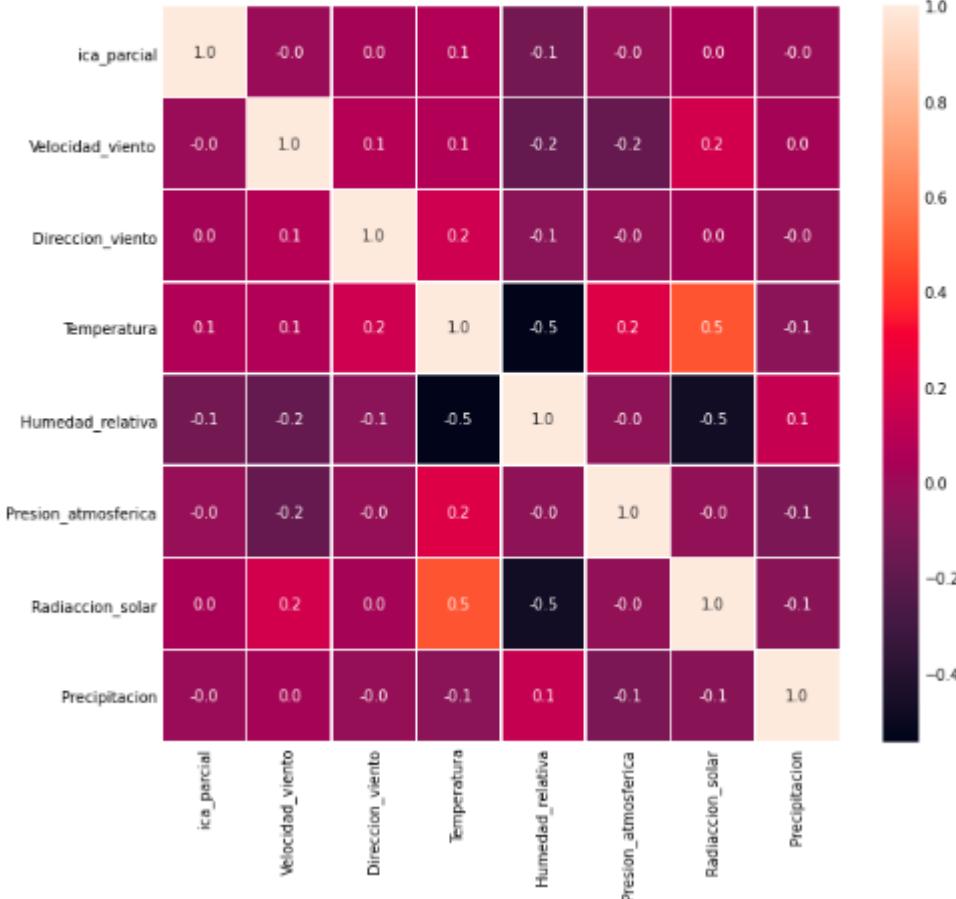


Ilustración 74. Matriz de correlación con datos

4.4.2. Desarrollo de los modelos

El dataframe usado para los algoritmos se llama datos_normalizados. Se incluye una columna nueva llamada ica de tipo binario (0,1) en caso de que supere el campo ica_parcial el valor de 60 se asigna 0, sino el valor es 1.

```
# Creamos campo binario ica para aquellos valores que superen percentil 75.
datos_filtrados['ica'] = np.where(datos_filtrados['ica_parcial']>=60, 0, 1) # Marcamos con 0 los que superen valor 60
datos_filtrados.head(5)
```

Creamos datos de entrenamiento y prueba con un porcentaje del 25% del total del conjunto de datos.

El dataset queda de la siguiente manera:

```
# Preparamos dataframe
datos_normalizados = datos_normalizados.drop(['fecha'], axis=1) # eliminamos fecha
datos_normalizados.insert(1, "ica", datos_filtrados["ica"]) # incluimos ica binario >60 1
datos_normalizados.head(5)

  ica ica_parcial Velocidad_viento Direccion_viento Temperatura Humedad_relativa Presion_atmosferica Radiacion_solar Precipitacion
0  1     0.000000      0.079137    0.041667    0.267241    0.849462    0.697872    0.008362      0.0
1  1     0.010472      0.079137    0.041667    0.267241    0.849462    0.697872    0.008362      0.0
2  1     0.014330      0.079137    0.041667    0.267241    0.849462    0.697872    0.008362      0.0
3  1     0.019485      0.079137    0.041667    0.267241    0.849462    0.697872    0.008362      0.0
4  1     0.082561      0.079137    0.041667    0.267241    0.849462    0.697872    0.008362      0.0

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import AdaBoostClassifier, GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score

X = datos_normalizados.drop('ica', axis=1)
y = datos_normalizados.ica

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
```

Ilustración 75. Dataset tras entrenamiento

4.4.2.1 Algoritmo K-NN

K Vecinos más Cercanos, KNN por sus siglas en inglés, es un algoritmo de aprendizaje de máquina muy simple, fácil de entender, versátil y uno de los más altos.

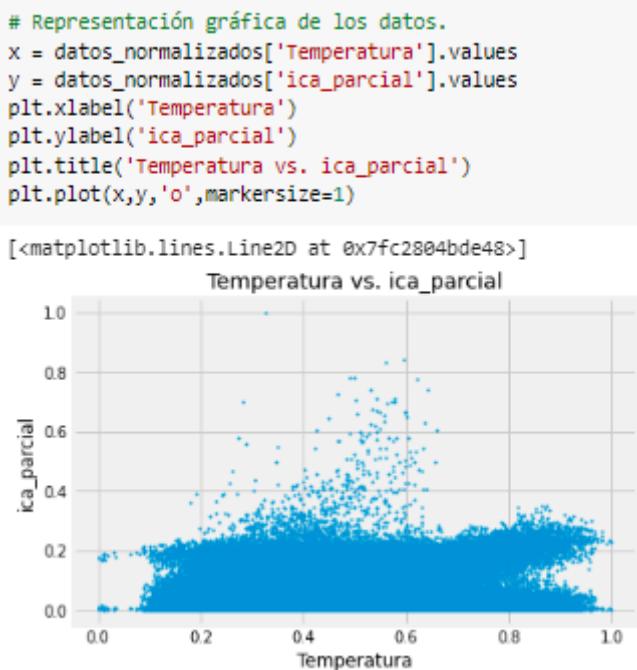


Ilustración 76. Temperatura vs ica_parcial - Algoritmo K-NN

El primer parámetro, por su puesto debe ser el del número de vecinos o K, aquí es donde defines este valor, a este parámetro se le conoce como n_neighbors. En nuestro caso lo hemos dejado por defecto a 5.

n_neighbors : int, optional (default = 5)

Number of neighbors to use by default for `kneighbors` queries.

Otro parámetro importante es definir la distancia que se utilizará para verificar los vecinos del dato que se está buscando predecir. Para configurar esto en el algoritmo se debe definir dos variables dentro del algoritmo, la primera es "p" y la segunda es "metric".

p : integer, optional (default = 2)

Power parameter for the Minkowski metric. When p = 1, this is equivalent to using manhattan_distance (l1), and euclidean_distance (l2) for p = 2. For arbitrary p, minkowski_distance (l_p) is used.

metric : string or callable, default 'minkowski'

the distance metric to use for the tree. The default metric is minkowski, and with p=2 is equivalent to the standard Euclidean metric. See the documentation of the DistanceMetric class for a list of available metrics.

"p" por defecto es igual a 2 y "metric" por defecto es "minkowski", con esta combinación se está eligiendo la distancia euclíadiana como la que se implementará.

```
knMod = KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,
                             metric='minkowski', metric_params=None)

knMod.fit(X_train, y_train)
knMod.score(X_test, y_test)

0.9963227144105259
```

Ilustración 77. Resultado K-NN

El resultado de este cálculo es de 0,9963227, este es un valor bastante aceptable.

4.4.2.2 Regresión Logística

La Regresión Logística es un método estadístico para predecir clases binarias. El resultado o variable objetivo es de naturaleza dicotómica. Dicotómica significa que solo hay dos clases posibles. Por ejemplo, se puede utilizar para problemas de detección de cáncer o calcular la probabilidad de que ocurra un evento, que es nuestro caso usando las magnitudes del objeto de estudio.

La Regresión Logística describe y estima la relación entre una variable binaria dependiente y las variables independientes.

```

glmMod = LogisticRegression(penalty='l1', dual=False, tol=0.0001, C=1.0, fit_intercept=True,
                            intercept_scaling=1, class_weight=None,
                            random_state=None, solver='liblinear', max_iter=100,
                            multi_class='ovr', verbose=2)

glmMod.fit(X_train, y_train)
glmMod.score(X_test, y_test)

[LibLinear]0.9998565597819709

test_labels=glmMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)

0.9999998751893564

```

Ilustración 78. Resultado Regresión Logística

El resultado de la precisión es de 0,999.

4.4.2.3 AdaBoost

AdaBoost, abreviatura de "Adaptive Boosting", es el primer algoritmo de impulso práctico propuesto por Freund y Schapire en 1996. Se centra en problemas de clasificación y tiene como objetivo convertir un conjunto de clasificadores débiles en uno fuerte. La ecuación final para la clasificación se puede representar como:

$$F(x) = \text{sign}\left(\sum_{m=1}^M \theta_m f_m(x)\right),$$

Ilustración 79. AdaBoost

donde f_m representa el clasificador débil m_{th} y θ_m es el peso correspondiente. Es exactamente la combinación ponderada de M clasificadores débiles. Todo el procedimiento del algoritmo AdaBoost se puede resumir de la siguiente manera.

Para el ajuste del modelo hemos configurados los parámetros de la siguiente forma:

- **base_estimator:** None (se usa para entrenar el modelo).
- **n_estimators:** 200 (número de estimadores para entrenar en cada iteración).
- **tasa_aprendizaje:** contribuye al peso de los estimadores. Utiliza 1 como valor predeterminado.

```
adaMod = AdaBoostClassifier(base_estimator=None, n_estimators=200, learning_rate=1.0)
```

```
adaMod.fit(X_train, y_train)
adaMod.score(X_test, y_test)
```

```
1.0
```

```
test_labels=adaMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)
```

```
1.0
```

Ilustración 80. Resultado AdaBoost

El resultado de la precisión es de 1.

4.4.2.4 GradientBoosting

Gradient Boosting Machine es un poderoso algoritmo de aprendizaje automático de conjunto que utiliza árboles de decisión.

El algoritmo es una generalización de AdaBoosting, que mejora el rendimiento del enfoque e introduce ideas de la agregación bootstrap para mejorar aún más los modelos, como muestrear aleatoriamente las muestras y las características cuando se ajustan los miembros del conjunto.

Hay muchos parámetros de configuración, a continuación, detallamos los valores predeterminados clave que hemos seleccionado en para nuestro proyecto.

- **tasa de aprendizaje** = 0.1 (contracción).
- **n_estimadores** = 200 (número de árboles).
- **max_depth** = 3.
- **min_samples_split** = 2.
- **min_samples_leaf** = 1.
- **submuestra** = 1.0.

```
gbMod = GradientBoostingClassifier(loss='deviance', learning_rate=0.1, n_estimators=200, subsample=1.0,
                                    min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
                                    max_depth=3,
                                    init=None, random_state=None, max_features=None, verbose=0)

gbMod.fit(X_train, y_train)
gbMod.score(X_test, y_test)

1.0

test_labels=gbMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)

1.0
```

Ilustración 81. Resultado GradientBoosting

El resultado de la precisión es de 1.

4.4.2.5 RandomForest

Es un tipo de algoritmo supervisado de aprendizaje automático basado en el aprendizaje conjunto.

Como con cualquier algoritmo, existen ventajas y desventajas de usarlo. Una desventaja importante de los árboles aleatorios radica en su complejidad. Requerían muchos más recursos computacionales, debido a la gran cantidad de árboles de decisión unidos.

```
rfMod = RandomForestClassifier(n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2,
                               min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto',
                               max_leaf_nodes=None, bootstrap=True, oob_score=False, n_jobs=1,
                               random_state=None, verbose=0)

rfMod.fit(X_train, y_train)
rfMod.score(X_test, y_test)

0.9999869599801792

test_labels=rfMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)

1.0
```

Ilustración 82. Resultado RandomForest

El resultado de la precisión es de 1.

4.4.3. Mejora de precisión del modelo

Nuestra elección en el análisis de la calidad del aire en la comunidad de Madrid, incluyendo la ciudad de Madrid, vino motivada por el periodo de

confinamiento que estábamos pasando en España a raíz del coronavirus. A priori teníamos la idea de que la calidad del aire al estar confinados sería mejor que en el periodo anterior y queríamos sacar un modelo que pudiera predecir cuando la calidad del aire empeoraría al salir del confinamiento y volver a la vida normal de contaminación por tráfico, fábricas, etc... A la vista de nuestro análisis esta primera hipótesis de mejora y posterior empeoramiento de la calidad del aire no se ha verificado, pues la calidad del aire en Madrid es buena, únicamente hay momentos puntuales , en horas y estaciones puntuales , donde se puede llegar a valores que dan una contaminación regular.

- Así pues primera conclusión es que la calidad del aire en Madrid en términos generales es tan buena que no empeora mucho ni mejora mucho con el confinamiento.
- De aquí también se puede sacar la conclusión de si la forma en la que se mide si una calidad del aire es la más correcta. Pues se realiza a través de un índice de calidad del aire (ICA) que marca el valor normalizado máximo de "alguno" de los 6 contaminantes que se miden en las estaciones de medición . Nos hace pensar si quizás sea más correcto calcular el índice de alguna otra forma con ponderaciones de determinados contaminantes en base a lo que perjudican unos u otros y realizando una mezcla de todos ellos sacar un índice más "real". Observamos que hay determinadas estaciones donde el ICA max de esa estación casi siempre lo da el mismo contaminante y es diferente contaminante en las estaciones de Madrid capital que en los pueblos de la comunidad.

Otra de nuestras suposiciones, a la hora de elegir nuestros datasets para el estudio, era predecir la calidad del aire mediante las mediciones metereológicas. Pensábamos que podían ser unos buenos predictores junto con el histórico de la calidad del aire.

- Lo que concluimos en esta parte es que el valor del índice de calidad del aire (ICA) no está nada correlacionado con las variables metereológicas. Así aunque alguna de las magnitudes de contaminaciones de forma independiente puede correlar un poco con las metereológicas , no llega a ser nada significativo.

Debido a que los datos del ICA de la calidad del aire, muestran sólo determinados momentos puntuales de baja calidad del aire, los modelos predictivos utilizados han aprendido más de los datos correspondientes a buena calidad del aire que a mala calidad.

Debido a ello, los modelos presentan métricas de performance muy buenas, no debido al modelo en sí, sino a la poca variación de los datos.

Es por ello por lo que los modelos predictivos se ajustan muy bien a los datos, no habiendo casi diferencias en términos de accuracy entre ellos.

4.5 Evaluación

4.6 Implementación

Para esta fase hemos decidido preparar un entorno de visualización apoyándonos en dockers, donde utilizando un contenedor ELK con ElasticSearch, Logstash y Kibana hemos podido cubrir con creces esta necesidad.

Se puede consultar el video explicativo donde se indica paso a paso la forma de proceder para dejar totalmente operativo el entorno y comenzar a trabajar sobre Kibana tras la inyección de datos en ElasticSearch desde Logstash.

En nuestro caso hemos utilizado el dataset con todos los datos recogidos por las estaciones y la ubicación de los mismos (longitud y latitud). La ubicación de las estaciones en longitud y latitud las hemos conseguido con un rpa de Uipath.

Muy a groso modo, los pasos seguidos hasta tener en disposición los datos en Kibana:

- Instalación de Docker Desktop
- Descarga de contenedor sebp/elk
- Arrancar el contendidor indicando los puertos para Elastic, Logstash y Kibana
- Copiar dataset al contendedor para procesarlo con Logstash
- Crear config para Logstash con nombre de index **global_info**

```
input {
    file {
        path => "/opt/kibana/src/plugins/home/server/services/sample_data/data_sets/calidadaire.csv"
        start_position => "beginning"
    }
}
filter {
    csv {
        columns => [ "id", "id_merge", "fechahora", "fecha", "hora", "estacion_real", "magnitud", "descripcion_magnitud", "factor_calculo_horario", "ica_parcial", "valor_magnitud", "provincia", "municipio", "dia_de_la_semana", "mag81_vel_viento", "mag82_dir_viento", "mag83_temperatura", "mag86_humedad", "mag87_presion_atm", "mag88_radiacionUV", "mag89_precipitacion", "ESTACION", "DIRECCION", "location" ]
        separator => ";"
    }
    date{
        match => ["fechahora","yyyy-M-d HH:mm"]
        target => "fechahora"
    }
}
output {
    elasticsearch {
        hosts => ["59f96e65d9bd:9200"]
        index => "global_info"
    }
}
```

Ilustración 83. Archivo de configuración para inyectar desde Logstash hacia elasticSearch

- Crear índice **global_info** con el mapeo del campo localización para que el tipo de dato sea **geo_point**

```

19 PUT global_info
20 {
21   "mappings": {
22     "properties": {
23       "location": {
24         "type": "geo_point"
25       }
26     }
27   }
28 }
```

Ilustración 84. Mapping para tipo geo_point en ElasticSearch

- Ejecutamos logstash para realizar carga con el comando: `logstash -f /opt/logstash/config/grupo7_BigData_global.config`

Y comprobamos que la carga se realizó correctamente con el comando : `curl http://localhost:9200/_cat/indices?v` donde podemos ver el número de líneas cargadas en el index y que coincide con el tamaño del archivo.

```

health status index      uuid                               pri rep docs.count docs.deleted store.size pri.store.size
yellow open  global_info S4Mn8gRERVqjzFKJxhAUZg    1   1      306745          0    164.8mb      164.8mb
green  open   .kibana_1  57koHSxmRkqwlboSRLRX7Q    1   0          9          1    30.8kb      30.8kb
```

Ilustración 85. Comprobación de carga en Elastic

- Una vez hecho esto debemos crear el indexPattern donde podremos ver que efectivamente disponemos de un campo fecha definido en el config más el campo geo_point definido en el mappings para su posterior representación :

Ilustración 86. Creación de indexPattern tras carga en Elastic

- Tras realizar esto ya podemos ir a la parte de Discover de Kibana donde veremos todos los datos cargados

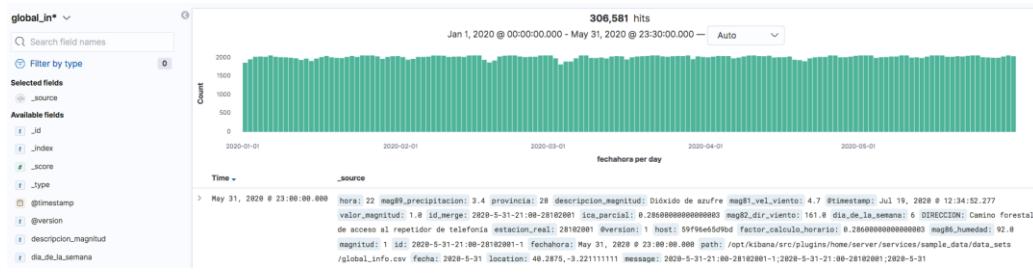


Ilustración 87. Discover en Kibana

Y a partir de aquí empezamos a presentar los datos que consideramos más significativos, en un primer dashboard se ha presentado la geolocalización de las estaciones de medición donde podremos observar las siguientes gráficas:

- En primer lugar vemos la ubicación de todas las estaciones utilizadas en el estudio donde por vemos que si queremos tener una imagen que abarque todos ellos se agrupan varias estaciones en los puntos centrales:

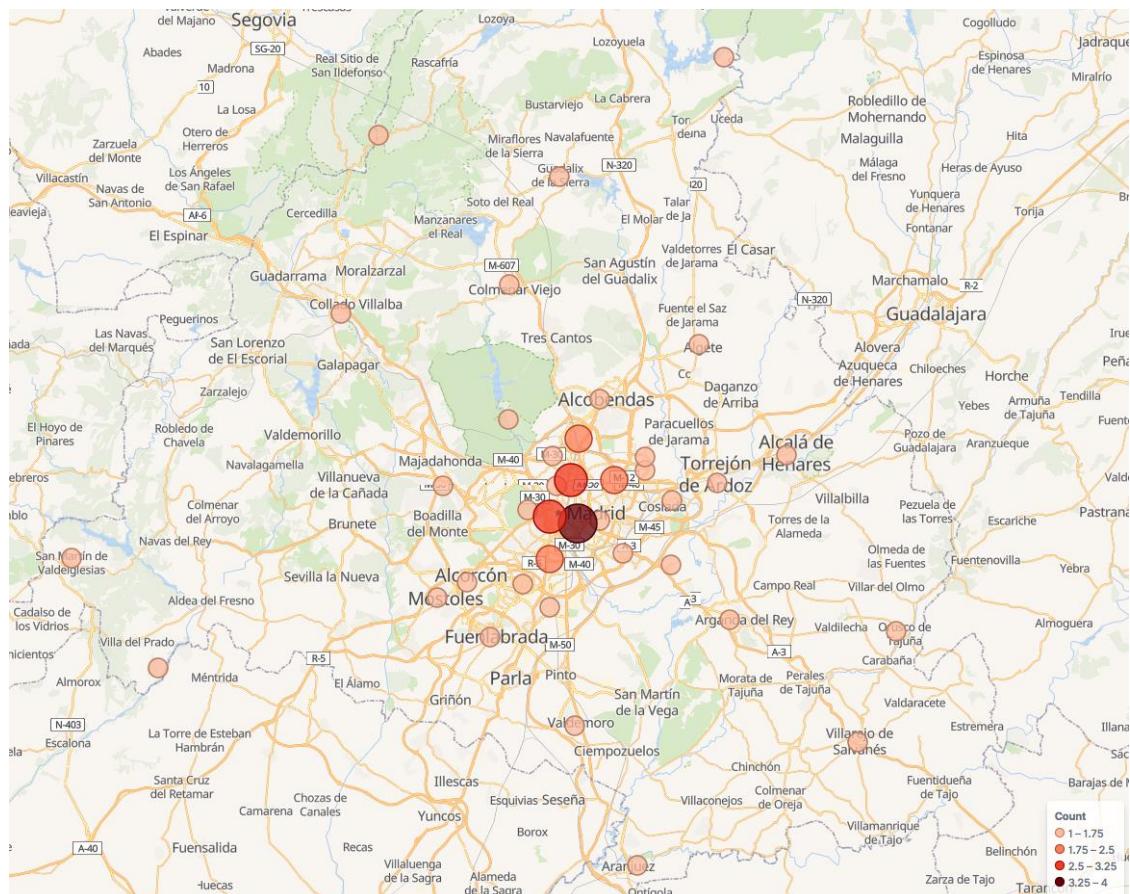


Ilustración 88. Mapa geolocalizado de las estaciones de medición de magnitudes

- Por otro lado vemos interesante mostrar el número de mediciones que ha realizado cada una de las estaciones entre los meses de enero y mayo donde podemos ver que hay ciertas estaciones que, bien no miden ciertas magnitudes o que por indisponibilidad no han estado el 100% del tiempo operativas:

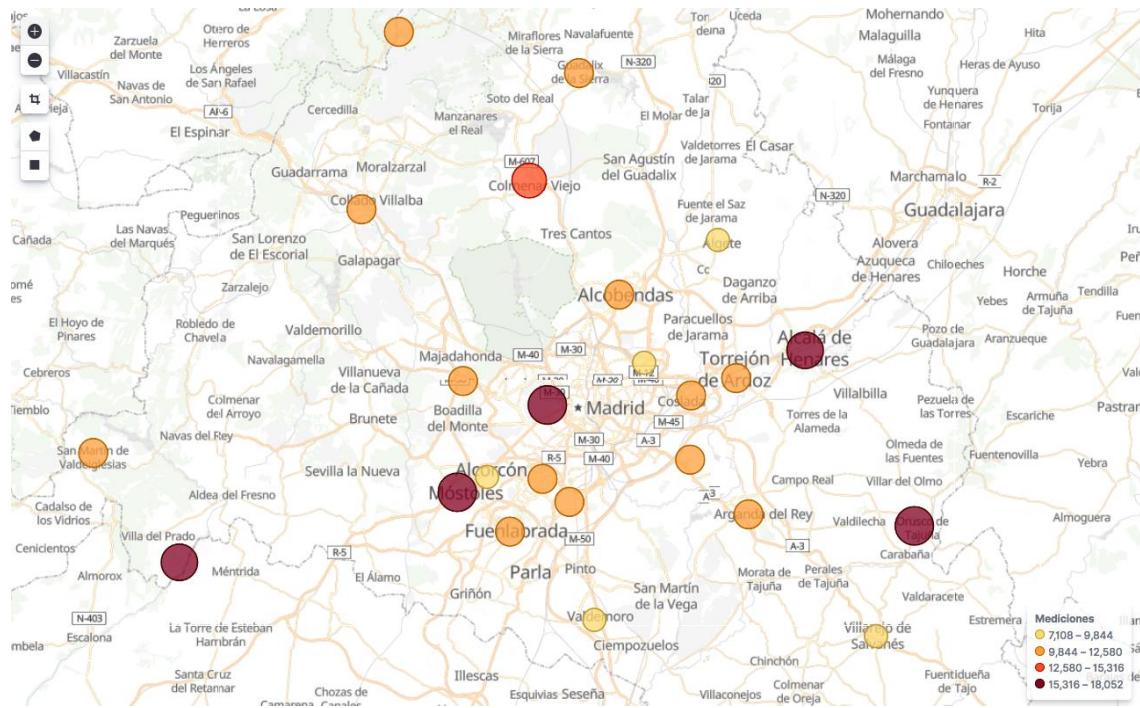


Ilustración 89. Mapa de geolocalización con el número de mediciones realizadas por cada estación en 5 meses.

Por otro lado, se ha creado otro dashboard donde usando el ica parcial de cada magnitud y con la función de tageo hemos obtenido las 1º mediciones más frecuentes por cada una de las magnitudes lo cual representaría un alto porcentaje de todas las mediciones realizadas :

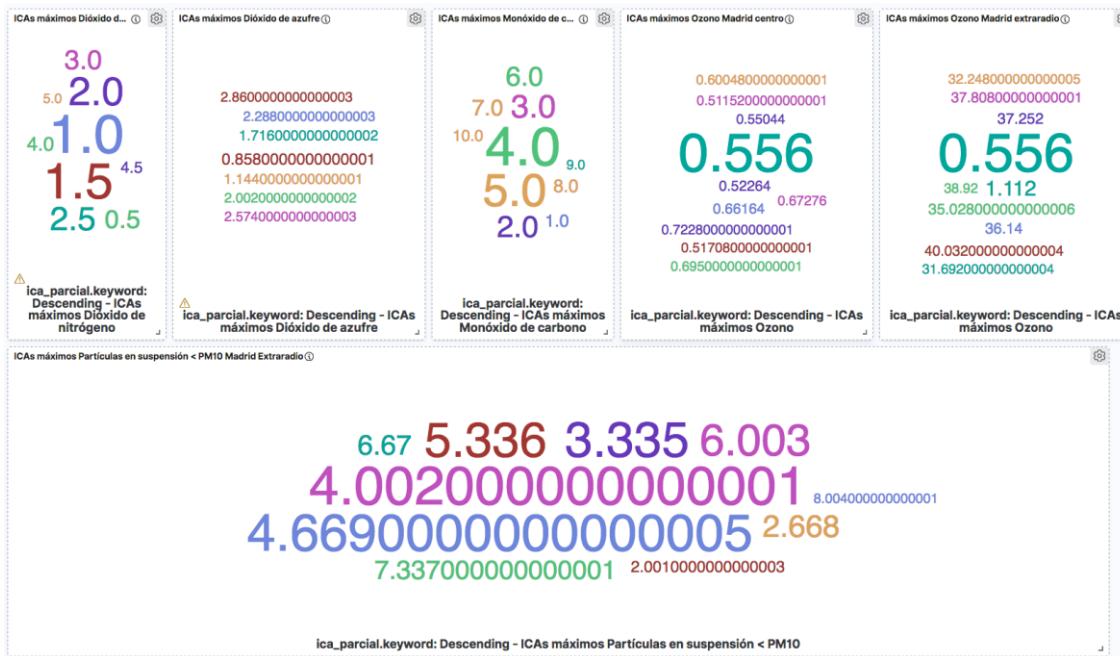


Ilustración 90. Dashboard en Kivana con las mediciones más frecuentes por magnitud contaminante.

Para conseguir esta representación se han creado filtros por magnitud por un lado y para el caso del ozono un filtro adicional para separar las mediciones de periferia y centro apoyándonos en el identificador único de cada estación.

5.- Conclusión

Dentro de este trabajo se han explorado la relación de las diferentes magnitudes en relación a la calidad del aire y su variación en el periodo con confinamiento decretado en España, y en particular en la Comunidad de Madrid desde el 14 de Marzo de 2020.

El presente análisis demuestra que las medidas de distanciamiento social impulsadas por las autoridades han tenido un efecto enorme en los niveles de contaminación atmosférica en Madrid, una ciudad que lleva años registrando datos de calidad de aire muy pobres y que el año pasado fue denunciada por la Comisión Europea por incumplir sistemáticamente los límites de NO₂ fijados por la normativa comunitaria, superior a los fijados para el límite de emisiones para grandes ciudades establecidos por La Directiva 2008/50/CE en una media anual de 40 µg/m³, en línea con las recomendaciones de la Organización Mundial de la Salud.

Pese a lo que pueda parecer a priori, las condiciones meteorológicas influyen mucho menos en la calidad del aire de lo que cabía esperar, no existiendo una fuerte relación entre ninguna de las magnitudes meteorológicas y las magnitudes de calidad del aire.

La situación creada por la pandemia de COVID-19 corrobora lo que la comunidad científica ha estado insistiendo durante años, que la reducción de las emisiones de tráfico en las ciudades tiene efectos claros en la reducción de la contaminación del aire, lo que representa una mejora significativa en la salud pública.

El análisis de las observaciones horarias de NO₂ en Madrid, indica una reducción promedio respectiva de 62%. Otro resultado destacado es que los valores pico máximos por hora también muestran reducciones significativas, con relaciones entre 1.2 y 1.7. La mejora en la calidad del aire ha ocurrido ampliamente, afectando tanto a los centros de las ciudades como a las áreas periféricas.

El periodo de confinamiento ha demostrado por tanto que el confinamiento de la sociedad ha influido positivamente en la calidad del aire, siendo (como es lógico) la acción social el verdadero impulsor de la contaminación ambiental. Y aunque se ha apreciado una disminución de monóxido de carbono a partir de marzo aunque no es la magnitud más próxima a convertirse en ICA en ninguna de las mediciones. Es por ello por lo que proponemos una revisión del cálculo del ICA, ya que su cálculo, tal y como hemos detallado en el documento está ligado a las magnitudes de calidad del aire y cada día se obtiene del valor mayor de cada una de esas magnitudes, por lo que no se debe usar para hacer estimaciones a futuro. Sería mas factible hacer una estimación de las magnitudes de calidad del aire y a partir de ahí, si se desea obtener el ICA que supuestamente se obtendría.

Teniendo la escala de ICA donde se indica que si el índice está por debajo de 75 la calidad del aire es buena y solo a partir de 100 es mala o muy mala concluimos que en la ventana temporal analizada entre enero y junio 2020 no ha habido ningún día que no haya sido buena. Por tanto, en la CAM generalmente la calidad del aire es buena.

Una estrategia integral para prevenir futuras epidemias similares a COVID-19 también debe diseñarse en términos de sostenibilidad, y no solo en relación con el sector de la salud. El vínculo es claro: la contaminación del aire es un factor de riesgo importante y contribuye al hecho de que las principales enfermedades crónicas aumentan su gravedad.

Estudio posterior:

A la vista de las conclusiones obtenidas en el trabajo, proponemos un estudio posterior sobre la composición del ICA.

El ICA debería estar conformado de forma ponderada por todos los agentes contaminantes del aire que se consideren (actualmente 6) por que, ¿Qué ocurre si el ICA parcial de cada uno de ellos quedara en 74? Estaríamos con un ICA indicando calidad del aire buena cuando todos ellos están en el umbral y muy probablemente el aire sea bastante perjudicial para la salud para cierto sector de la población.

6.- Anexos

6.1 ANEXO I. CÓDIGOS DE ESTACIONES

Las estaciones señaladas con un asterisco (*), cambiaron su código a partir de la fecha que se indica para la adaptación a la codificación nacional de intercambio de datos de calidad del aire.

28079001	Pº. Recoletos	Baja.- 04/05/2009 (14:00 h.)
28079002	Gita. de Carlos V	Baja.- 04/12/2006 (11:00 h.)
28079003 28079035(*)	Pza. del Carmen	* Código desde enero 2011
28079004	Pza. de España	
28079005 28079039(*)	Barrio del Pilar	* Código desde enero 2011
28079006	Pza. Dr. Marañón	Baja.- 27/11/2009 (08:00 h.)
28079007	Pza. M. de Salamanca	Baja.- 30/12/2009 (14:00 h.)
28079008	Escuelas Aguirre	
28079009	Pza. Luca de Tena	Baja.- 07/12/2009 (08:00 h.)
28079010 28079038(*)	Cuatro Caminos	* Código desde enero 2011
28079011	Av. Ramón y Cajal	
28079012	Pza. Manuel Becerra	Baja.- 30/12/2009 (14:00 h.)
28079013 28079040(*)	Vallecas	* Código desde enero 2011
28079014	Pza. Fdez. Ladreda	Baja.- 02/12/2009 (09:00 h.)
28079015	Pza. Castilla	Baja.- 17/10/2008 (11:00 h.)
28079016	Arturo Soria	
28079017	Villaverde Alto	
28079018	C/ Farolillo	
28079019	Huerta Castañeda	Baja.- 30/12/2009 (13:00 h.)
28079020 28079036(*)	Moratalaz	* Código desde enero 2011
28079021	Pza. Cristo Rey	Baja.- 04/12/2009 (14:00 h.)
28079022	Pº. Pontones	Baja.- 20/11/2009 (10:00 h.)
28079023	Final C/ Alcalá	Baja.- 30/12/2009 (14:00 h.)
28079024	Casa de Campo	
28079025	Santa Eugenia	Baja.- 16/11/2009 (10:00 h.)
28079026	Urb. Embajada (Barajas)	Baja.- 11/01/2010 (09:00 h.)
28079027	Barajas	
28079047	Méndez Álvaro	Alta.- 21/12/2009 (00:00 h.)
28079048	Pº. Castellana	Alta.- 01/06/2010 (00:00 h.)
28079049	Retiro	Alta.- 01/01/2010 (00:00 h.)
28079050	Pza. Castilla	Alta.- 08/02/2010 (00:00 h.)
28079054	Ensanche Vallecas	Alta.- 11/12/2009 (00:00 h.)
28079055	Urb. Embajada (Barajas)	Alta.- 20/01/2010 (15:00 h.)
28079056	Plaza Elíptica	Alta.- 18/01/2010 (12:00 h.)
28079057	Sanchinarro	Alta.- 24/11/2009 (00:00 h.)
28079058	El Pardo	Alta.- 30/11/2009 (13:00 h.)
28079059	Parque Juan Carlos I	Alta.- 14/12/2009 (00:00 h.)
28079086 28079060(*)	Tres Olivos	Alta.- 14/01/2010 (13:00 h.) * Código desde enero 2011

6.2 ANEXO II. MAGNITUDES, UNIDADES Y TÉCNICAS DE MEDIDA

Magnitud		Abreviatura o fórmula	Unidad medida	Técnica de medida	
01	Dióxido de Azufre		µg/m ³	38	Fluorescencia ultravioleta
06	Monóxido de Carbono	SO ₂ CO	mg/m ³	48	Absorción infrarroja
07	Monóxido de Nitrógeno	NO	µg/m ³	08	Quimioluminiscencia
08	Dióxido de Nitrógeno	NO ₂	µg/m ³	08	Id.
09	Partículas < 2.5 µm	PM2.5	µg/m ³	47	Microbalanza
10	Partículas < 10 µm	PM10	µg/m ³	47	Id.
12	Óxidos de Nitrógeno	NOx	µg/m ³	08	Quimioluminiscencia
14	Ozono	O ₃	µg/m ³	06	Absorción ultravioleta
20	Tolueno	TOL	µg/m ³	59	Cromatografía de gases
30	Benceno	BEN	µg/m ³	59	Id.
35	Etilbenceno	EBE	µg/m ³	59	Id.
37	Metaxileno	MXY	µg/m ³	59	Id.
38	Paraxileno	PXY	µg/m ³	59	Id.
39	Ortoxileno	OXY	µg/m ³	59	Id.
42	Hidrocarburos totales (hexano)	TCH	mg/m ³	02	Ionización de llama
43	Metano	CH4	mg/m ³	02	Id.
44	Hidrocarburos no metánicos (hexano)	NMHC	mg/m ³	02	Id.

6.3 ANEXO III. VALORES LÍMITE

Compuesto	Valor límite / objetivo / Umbral de Alerta	Concentración	Nº máximo de superaciones
PM₁₀	Media anual. Media diaria.	40 µg/m ³ 50 µg/m ³	35 días/año
PM_{2,5}	Media anual.	25 µg/m ³	
SO₂	Media diaria. Media horaria. Umbral de alerta (3 horas consecutivas en área representativa de 100 km o zona o aglomeración entera).	125 µg/m ³ 350 µg/m ³ 500 µg/m ³	3 días/año 24 horas/año
NO₂	Media anual. Media horaria. Umbral de alerta (3 horas consecutivas en área representativa de 100 km o zona o aglomeración entera).	40 µg/m ³ 200 µg/m ³ 400 µg/m ³	18 horas /año
Pb	Media anual.	0,5 µg/m ³	
CO	Máxima diaria de las medias móviles octohorarias.	10 mg/ m ³	
C₆H₆	Media anual.	5 µg/m ³	
O₃	Máxima diaria de las medias móviles octohorarias. Umbral de información. Media horaria. Umbral de alerta. Media horaria.	120 µg/m ³ 180 µg/m ³ 240 µg/m ³	25 días /año, promediados en un período de 3 años
Arsénico	Media anual.	6 ng/ m ³	
Cadmio	Media anual.	5 ng/ m ³	
Niquel	Media anual.	20 ng/ m ³	
Benzo (a) pireno	Media anual.	1 ng/ m ³	

Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire.

6.4 ANEXO IV. CÓDIGO PYTHON API AEMET

```

# Importación de paquetes necesarios
# En caso de no tener algún paquete instalado habrá que instalarlo. Se puede hacer desde
CMD, por ejemplo: python -m pip install unicodecsv
import requests
import json
import re
import csv
import unicodecsv as csv
import itertools
import calendar, datetime
from datetime import date, timedelta
import time
# Obviamos aviso de HTTPS
from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
# Definimos los datos de cabecera + API_KEY recibida para hacer la llamada HTTP
proxies = {
    'http': 'http://180.185.219.10:8080',
    'https': 'http://180.185.219.10:8080',
}
api_key = {
    "api_key": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIoIjNzWx1Y2FsemFkYUBnbWFpbC5jb20iLCJqdGkiOi5Zjc3NmYxNy1jNm
VlTQzZWYtOTJjMy1lYzhZWUyNWUyOWEiLCJpc3MiOiJBRU1FVCIsImhdCl6MTUzNzk5Mzl2MCwidXNlcklkj
oiOWY3NzMTCtYzIZS00M2VmLTkyYzMtZWM4YWVIMjVIMjlhiwicm9sZSI6IiJ9.6GfoHC3sElRnJcE_OtZRg
ui8wip1cgp8OSgBwl3EBMs"
cabecera_llamada = {'Accept':
    "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
    'Accept-Encoding': "gzip, deflate, sdch, br",
    'Accept-Language': "es-ES,es;q=0.8,en;q=0.6",
    'Cache-Control': "max-age=0",
    'Connection': "keep-alive",
    'Host': "opendata.aemet.es",
    'Upgrade-Insecure-Requests': "1",
    'User-Agent': "Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.76 Safari/537.36"}}

# Loop para crear la lista de primeros días y últimos días del mes(es un loop anidado)
inicio = 2020
fin = 2020
# Creamos dos listas iniciales vacías y luego se alimentan
# Lista con día de inicio de mes
# Lista con fin de mes
# Posteriormente las combinaremos creando tuplas
mesesinicio = []
mesesfinal = []
for i in range(inicio, fin + 1):
    for j in range(1, 3):
        primerdia = datetime.date(i, j, 1)
        ultimodia = calendar.monthrange(i, j)
        ultimodia2 = datetime.date(i, j, ultimodia[1])
        mesesinicio.append(primerdia)
        mesesfinal.append(ultimodia2)
        print(primerdia.strftime('%m-%d-%Y') + ' ' + ultimodia2.strftime('%m-%d-%Y'))
        print("\n")
# Utilizamos la función ZIP para combinar ambas listas y crear una lista de tuplas con
# día inicio - fin mes
listameses = list(zip(mesesinicio,mesesfinal))

# Listado de estaciones
#
# 3100B - Aranjuez
# 3110C - Buitrago Del Lozoya
# 3191E - Colmenar Viejo
# 3200 - Getafe
# 3129 - Madrid Aeropuerto
# 3194U - Madrid, Ciudad Universitaria
# 3196 - Madrid, Cuatro Vientos
# 3195 - Madrid, Retiro
# 3266A - Puerto Alto Del Leon

```

```
# 2462 - Puerto de Navacerrada
# 3338 - Robledo De Chavela
# 3111D - Somosierra
# 3175 - Torrejon De Ardoz
estaciones = ['3100B', '3110C', '3191E', '3200', '3129', '3194U', '3196', '3195',
'3266A', '2462', '3338', '3111D', '3175']
# Ahora vamos a crear las posibles combinaciones de Día inicio - Día fin mes - Estación
de medición
# Para ello combinamos la lista de tuplas de fechas que tenemos con la lista de
estaciones
combinaciones = list(itertools.product(listameses, estaciones))
# Defino la lista de campos en función de los metadatos proporcionados por la web de
AEMET
# Esta listado se utilizará como cabecera del CSV que se genere
campos = ['fecha', 'indicativo', 'nombre', 'provincia', 'latitud', 'lmed', 'prec',
'tmin', 'horatmin', 'tmax', 'horatmax', 'dir', 'velmedia', 'racha', 'horaracha', 'sol',
'presMax', 'horaPresMax', 'presMin', 'horaPresMin']
# Loop de consultas a la web de AEMET
for i in combinaciones:
    url =
        "https://opendata.aemet.es/opendata/api/valores/climatologicos/diarios/datos/fechaini/{fechainicio}T00:00:00UTC/fechafin/{fechafin}T23:59:59UTC/estacion/{idestacion}/"
    # Con la funcion lista.index() conseguimos el índice de la lista que sirve para
    obtener cada una de las fechas en cada iteración
    # [0] Corresponde a primer día de mes
    fechainicio = combinaciones[combinaciones.index(i)][0][0].strftime('%Y-%m-%d')
    # [1] Corresponde a último día de mes
    fechafin = combinaciones[combinaciones.index(i)][0][1].strftime('%Y-%m-%d')
    # Marcamos con [1] para quedarnos con la segunda parte de la tupla
    estacionreemplazo = combinaciones[combinaciones.index(i)][1]
    # Reemplazo fecha inicio
    urlaux = re.sub(r'{fechainicio}', fechainicio, url)
    # Remplazo fecha fin sobre urlaux anterior
    urlaux2 = re.sub(r'{fechafin}', fechafin, urlaux)
    # Reemplazo idestacion sobre urlaux2 anterior
    urlnueva = re.sub(r'{idestacion}', estacionreemplazo, urlaux2)
    print("Descargando" + " " + urlnueva)
    respuesta = requests.get(urlnueva, params=api_key, headers=cabecera_llamada,
    verify=False)
    # AEMET devuelve un enlace temporal que contiene los datos en formato JSON
    json_data = json.loads(respuesta.text)
    # Parseamos el fichero JSON extrayendo el link a la web
    urldatos = json_data['datos']
    # Hacemos una nueva llamada a la web temporal que tiene los datos
    respuesta2 = requests.get(urldatos, params=api_key, headers=cabecera_llamada,
    verify=False)
    json_data2 = json.loads(respuesta2.text)
```

6.5 INFORMACION RELEVANTE DEL TRABAJO GRUPAL

Efectos del confinamiento en la calidad del aire de la Comunidad de Madrid

Este proyecto es una práctica docente. Su objetivo es poner en práctica las técnicas de Big Data y Machine Learning aprendidas.

Este documento forma parte de un proyecto publicado en <https://github.com/Big-Data-Equipo-7/Proyecto>

Integrantes del Grupo 7

- Alfonso Gallardo (Científico de datos)
- Raúl Hervás (Analista de datos)
- Carmen Reina (Analista de negocio)
- Walter Ronceros (Arquitecto de datos)
- Susana Vara (Analista de datos - Representante)

Limitaciones

Para este estudio damos por hecho ciertas limitaciones insalvables como son:

- No todas las estaciones contienen medición de todos los agentes contaminantes.
- Las mediciones meteorológicas no se pueden prever.
- El índice ICA (Índice Calidad del Aire) se obtiene de la medición más adversa de 5 agentes contaminantes por lo que no puede utilizarse para buscar correlaciones directas.

Tecnologías

- Python 3.8.2
- Anaconda Navigator 3.0.1
- Docker 19.03.8
- Jupyter Notebook 6.0.3
- Spyder
- Elasticsearch
- Databricks

- Google Colab

Instrucciones para la reproducción del trabajo

A tener en cuenta: Los scripts están preparados para funcionar en rutas concretas y han de ejecutarse en el mismo orden para tener los resultados esperados.

1. Ejecución del script 1 - *Generar datos de Madrid desagregado.py* para obtener el dataset que se utilizará en el siguiente punto (datosdefinitivos.csv).
2. Ejecución del [Notebook albergado en Colab](#). Se guarda una copia dentro del proyecto con el nombre 2 - *Estudio datos de Madrid.ipynb*. Hay que tener en cuenta de que la reproducción de los modelos están preparadas para funcionar en Google Colab, la ejecución en local requiere algunas modificaciones del Notebook.
3. Ejecución del script 3 - *Generar datosHistoricosCalidadDelAire.py* para obtener el dataset que se utilizará en el siguiente punto (datosHistoricosCalidadAire.csv)
4. Ejecución del Notebook Jupyter 4 - *Estudio datosHistoricosCalidadDelAire.ipynb*

Instrucciones para entorno de visualización

Esta fase se realiza apoyándonos en dockers donde utilizaremos la pila **ELK**.

Podrás ver un video explicativo en el siguiente enlace: [Video explicativo](#) (*Descargar el vídeo para una mayor calidad de visualización*)

Necesitarás

- Instalar [Docker Desktop](#)
- Descarga de contenedor [sebp/elk](#)
- Arrancar el contenedor indicando los puertos para Elastic, Logstash y Kibana
- Copiar dataset al contenedor para procesarlo con Logstash
- Crear config para Logstash con nombre de index global_info
- Crear índice global_info con el mapeo del campo localización para que el tipo de dato sea geo_point
- Ejecutamos logstash para realizar carga con el comando: logstash -f /opt/logstash/config/grupo7_BigData_global.config

Para mas detalle consultar la memoria y/o ver el [video explicativo](#).

Índice de links externos

- Proyecto en Github: <https://github.com/Big-Data-Equipo-7/Proyecto>
- Notebook: [Colab de Google](#)
- Video demostrativo de Docker y ELK: [Video](#) (*Descargar el vídeo para una mayor calidad de visualización*)
- Dataset Datos Meteorológicos Comunidad de Madrid: [Dataset](#)
- Dataset Datos Calidad del Aire Comunidad de Madrid: [Dataste](#)
- Dataset Estaciones Comunidad de Madrid: [Dataset](#)
- Dataste Datos Meteorológicos Ayuntamiento de Madrid: [Dataset](#)
- Dataste Datos Calidad del Aire Ayuntamiento de Madrid: [Dataset](#)
- Dataset Estaciones Ayuntamiento de Madrid: [Dataset](#)
- Descarga de software: [Docker Desktop](#)
- Descarga de contenedor Docker: [sebp/elk](#)

7.- Tabla de ilustraciones

Ilustración 9. Entorno de trabajo en Trello.....	17
Ilustración 10. Desarrollando en Google Colab.	15
Ilustración 11. Proyecto en GitHub.....	16
Ilustración 1. Evolución emisiones dióxidos de nitrógeno UE28. (Fuente: Agencia Europea del Medio Ambiente)	23
Ilustración 2. Esquema de formación de lluvia ácida.....	24
Ilustración 3. Estaciones calidad aire por zona. (Fuente: Ayto. Madrid - Protocolo de actuación para episodios de contaminación por dióxido de nitrógeno)	27
Ilustración 4. Delimitación de zonas a efectos de aplicación del protocolo de actuación para episodios de contaminación por dióxido de nitrógeno. (Fuente: Ayto. Madrid).....	27
Ilustración 5. Casos diarios confirmados de COVID-10 por fecha de inicio de síntomas y de diagnóstico desde el 11/05/2020 por Comunidades Autónomas, a 19/06/2020.....	29
Ilustración 6. Número de fallecidos diario por COVID-19 por fecha de defunción en España a 18/06/2020	30
Ilustración 7. Casos de COVID-19 que han precisado hospitalización, ingreso en UCI y fallecidos por Comunidades Autónomas en España a 19/06/2020	31
Ilustración 8. Cronograma de Confinamiento, Desescalada y Nueva Normalidad.....	33
Ilustración 12. Muestra de formato de dataset original	34
Ilustración 13. Muestra de códigos de estaciones.....	34
Ilustración 14. Muestra de Municipios incluidos en el estudio	35
Ilustración 15. Las magnitudes de calidad del aire y las técnicas de medida	35
Ilustración 16. Las magnitudes meteorológicas en las estaciones	35
Ilustración 17. Captura de pantalla de AQICN.ORG.....	36
Ilustración 18. Niveles de calidad del aire	36
Ilustración 19. Ejemplo de código para el cálculo de ICA parcial	38
Ilustración 20. Contaminantes y factores de cálculo para la obtención del ICA	38
Ilustración 21. Captura de portal del ayuntamiento de Madrid (Calidad del aire).....	39
Ilustración 22. Captura de portal de la Comunidad de Madrid (Calidad el aire)	39
Ilustración 23. Captura de portal del Ayuntamiento de Madrid (Datos meteorológicos)	40
Ilustración 24. Ejemplo bucle para desagrupar horas	42
Ilustración 25. Unión con las variables meteorológicas.....	42
Ilustración 26. Creamos los dataframe.....	42
Ilustración 27. Carga de datos y librerías en google.colab	43
Ilustración 28. Importación de datos en google.colab	43
Ilustración 29. Visualización de datos en google.colab	43
Ilustración 30. Dataset. Tipo de datos.....	44

Ilustración 30. ICA. Tipo de datos de cada columna	45
Ilustración 30. ICA. Comprobaciones datos	45
Ilustración 30. ICA. Filtrado ICA distinto 0	46
Ilustración 30. ICA. Suma de ICA Hora/magnitudes en 24 horas	47
Ilustración 30. ICA. Mediciones magnitudes.....	47
Ilustración 30. ICA. Ozono estable	48
Ilustración 30. ICA. Suma de ICA Hora/magnitudes 2018.....	49
Ilustración 30. ICA. Suma de ICA Hora/magnitudes 2019	49
Ilustración 30. ICA. Suma de ICA Hora/magnitudes 2020	50
Ilustración 30. ICA. Concentración valores y outliers.....	51
Ilustración 31. ICA. Asimetría y curtosis	51
Ilustración 32. ICA. Distribución valores ICA.....	52
Ilustración 33. ICA. Grafica violín.....	52
Ilustración 30. Media mensual ICA.....	53
Ilustración 30. Promedio ICA mes	53
Ilustración 30. Distribución ICA horaria	54
Ilustración 30. Velocidad del viento. Histograma	55
Ilustración 31. Velocidad del viento. Asimetría y curtosis.....	55
Ilustración 32. Velocidad del viento. Concentración y outliers.....	56
Ilustración 33. Velocidad del viento. Distribución magnitud	56
Ilustración 34. Velocidad del viento. Valores magnitud	57
Ilustración 30. Dirección del viento. Histograma	58
Ilustración 31. Dirección del viento. Asimetría y curtosis.....	58
Ilustración 32. Dirección del viento. Concentración y outliers	59
Ilustración 33. Dirección del viento. Conversión fecha dataframe	59
Ilustración 30. Temperatura. Histograma.....	60
Ilustración 31. Temperatura. Gráfico violín	60
Ilustración 32. Temperatura. Concentración y outliers	61
Ilustración 30. Humedad relativa. Histograma	61
Ilustración 31. Humedad relativa. Gráfico violín	62
Ilustración 32. Humedad relativa. Concentración y outliers	62
Ilustración 30. Presión atmosférica. Histograma	63
Ilustración 31. Presión atmosférica. Asimetría y curtosis	63
Ilustración 32. Presión atmosférica. Concentración y outliers	64
Ilustración 30. Radiación solar. Histograma	64
Ilustración 31. Radiación solar. Asimetría y curtosis.....	65
Ilustración 32. Radiación solar. Concentración y outliers.....	65
Ilustración 30. Precipitación. Histograma.....	66
Ilustración 31. Precipitación. Asimetría y curtosis	66
Ilustración 32. Precipitación. Concentración de valores y outliers	67
Ilustración 30. Preparación de modelado	67
Ilustración 30. Diagrama box y whisker.....	68
Ilustración 30. Matriz de correlación	69
Ilustración 31. Matriz de correlación con datos	70
Ilustración 30. Dataset tras entrenamiento	71
Ilustración 31. Temperatura vs ica_parcial - Algoritmo K-NN	71

Ilustración 32. Resultado K-NN.....	72
Ilustración 32. Resultado Regresión Logística.....	73
Ilustración 33. AdaBoost.....	73
Ilustración 34. Resultado AdaBoost	74
Ilustración 35. Resultado GradientBoosting	75
Ilustración 36. Resultado RandomForest	75
Ilustración 30. Archivo de configuración para inyectar desde Logstash hacia elasticSearch.....	77
Ilustración 31. Mapping para tipo geo_point en ElasticSearch	78
Ilustración 32. Comprobación de carga en Elastich	78
Ilustración 33. Creación de indexPattern tras carga en Elastic	78
Ilustración 34. Discover en Kibana	79
Ilustración 35. Mapa geolocalizado de las estaciones de medición de magnitudes.....	79
Ilustración 36. Mapa de geolocalización con el número de mediciones realizadas por cada estación en 5 meses.....	80
Ilustración 37. Dashboard en Kivana con las mediciones más frecuentes por magnitud contaminante.	81

8.- Bibliografía

1. Aqicn.org. -> <http://aqicn.org/city/madrid/es/>
2. Calidad del aire Ayuntamiento de Madrid -> <https://datos.madrid.es/>
3. Calidad del aire Comunidad de Madrid ->
<http://datos.comunidad.madrid/>
4. Datos meteorológicos Ayuntamiento de Madrid ->
<https://datos.madrid.es/>
5. Python: <https://docs.python.org/3/reference/>
6. Pandas: <https://pandas.pydata.org/docs/reference/index.html>
7. Seaborn: <https://seaborn.pydata.org/>
8. Matplotlib: <https://matplotlib.org>
9. Protocolo de actuación para episodios de contaminación por dióxido de nitrógeno.
10. Memoria Calidad del Aire 2017 elaborada por el Ayuntamiento de Madrid.
3. Agencia Europea del Medio Ambiente.
4. D.Michie, D.J.Spiegelhalter, C.C.Taylor "Machine Learning, Neural and Statistical Classification", 1994.
5. Breiman, L. (1996) "Bagging predictors". Machine Learning,.
6. Breiman, L. (2001) "Random forests". Machine Learning,
7. Cuadras, C.M. (2014). "Nuevos Métodos de Análisis Multivariante." 8. Portela, J. Macros validación cruzada para las distintas técnicas de modelización.
8. Athira, V^a; Geetha P; Vinayakumar, Rab; Soman, K P (2018). DeepAirNet: Applying Recurrent Networks for Air Quality Prediction. - Averett, Nancy (2015). Air Pollution and Birth Weight: New Clues about a Potential Critical Window of Exposure.
9. Ayuntamiento de Madrid (2020). Protocolo de actuación para episodios de contaminación por dióxido de nitrógeno.
10. A systematic review of data mining and machine learning for air pollution epidemiology.
11. Gu, Jifa; Zhang, Lingling (2014). Data, DIKW, Big data and Data science.
12. IBM (Sin año). IBM SPSS Modeler CRISP-DM Guide.
13. Li, Xiang; Shao, Jing; Hu, Yuan (2016). Deep learning arquitecture for air quality predictions.
14. Organización Mundial de la Salud (OMS) (2018). Contaminación del aire de interiores y salud.
15. Organización Mundial de la Salud (OMS) (2018). Contaminación atmosférica y salud infantil.
16. Pardo, Malpica (2017). Air quality forecasting in Madrid using Long Short-Term Memory
17. Rodríguez Miranda, Alejandro Aurelio (2018). Modelización y análisis de la calidad del aire en la ciudad de Oviedo (norte de España),

- mediante los enfoques PSO-SVM, red neuronal MLP y árbol de regresión M5.
- 18. Díaz, Julio; Linares, Cristina (2010). Las causas de la contaminación y los contaminantes atmosféricos más importantes.
 - 19. Julià Minguillón (2016). Fundamentos de data science.
 - 20. Li, Xiang; Shao, Jing; Hu, Yuan (2016). Deep learning arquitecture for air quality predictions.

9. Glosario

A continuación, se encuentra la definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria:

- AEMET: Agencia Estatal del Meteorología
- API Rest: Application Programming Interface Representational State Transfer
- Clustering: Algoritmo de agrupamiento
- CMAQ: Community Multi-Scale Air Quality
- CNN: Convolutional Neural Network
- CO: Monóxido de Carbono
- CO₂: Dióxido de Carbono
- COV: Compuestos Orgánicos Volátiles
- COVNM: Compuestos Orgánicos Volátiles No Metálicos
- CSV: Comma Separated Values
- Dataframe: Objeto usado para almacenar tablas de datos
- Hadoop: Framework de software para aplicaciones distribuidas
- HDFS: Hadoop Distributed File System
- Hive: Infraestructura de almacenamiento de datos distribuido con lenguaje HSQL
- Github: Plataforma de desarrollo corporativo que utiliza el control de versiones Git
- GPL: General Public License
- ICA: Índice de calidad del aire
- K-means: Algoritmo de agrupamiento (clustering)
- MVS: Máquinas de Vector Soporte
- MXy: Metaxileno
- NO: Monóxido de Nitrógeno
- NO₂: Dióxido de Nitrógeno
- NOX: Óxidos de Nitrógeno
- O₃: Ozono
- OpenData: Datos abiertos disponibles públicamente sin restricciones
- 73
- PM_{0,1}: Partículas ultrafinas o de diámetro aerodinámico $\leq 0,1 \mu\text{m}$
- PM_{2,5}: Partículas finas o de diámetro aerodinámico $\leq 2,5 \mu\text{m}$
- PM₁₀: Partículas gruesas o de diámetro aerodinámico $\leq 10 \mu\text{m}$
- PMx: Partículas subméticas o en suspensión (diámetro $\leq 10 \mu\text{m}$)
- SVM: Support Vector Machines
- SVR: Support Vector Regression
- SO₂: Dióxido de Azufre
- SOX: Óxidos de Azufre