UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Geodésica, Cartográfica y Topográfica

Descarga, tratamiento y análisis de las mediciones de contaminantes atmosféricos en la Comunidad Valenciana.

Trabajo Fin de Grado

Grado en Ingeniería Geomática y Topografía

AUTOR/A: Simarro González, Adrián

Tutor/a: Coll Aliaga, Peregrina Eloína

Cotutor/a: Porres de la Haza, Maria Joaquina

Director/a Experimental: LORENZO SAEZ, EDGAR

CURSO ACADÉMICO: 2021/2022

AGRADECIMIENTOS

Este trabajo no hubiera sido posible sin el apoyo incondicional de mis padres, sin las enseñanzas de mis tutores y sin el respaldo de mis compañeros. Gracias a la Cátedra de Gobernanza de la UPV, especialmente a Eloína Coll Aliaga y Edgar Lorenzo Sáez por brindarme la oportunidad de participar en este proyecto.

Quiero hacer una mención especial a Adrián Portales Girona, Javier Solá Ferrer y María Joaquina Porres de la Haza con los que he trabajado codo con codo en el Grupo de Calidad del Aire, en el que hemos demostrado una gran cooperación y confianza entre nosotros.

COMPROMISO

"El presente documento ha sido realizado completamente por el firmante; no ha sido entregado como otro trabajo académico previo y todo el material tomado de otras fuentes ha sido convenientemente entrecomillado y citado su origen en el texto, así como referenciado en la bibliografía"

RESUM

L'objectiu del projecte és la creació de metodologies de descàrrega massiva de dades com estacions meteorològiques a terra i imatges satel·litàries Sentinel 5P de diferents productes, aplicant-se un tractament que regeixi ordre i qualitat entre les dades. Per això s'usarà Python per a la creació dels codis de descàrrega mitjançant la implementació de diverses plataformes com pot ser Google Earth Engine juntament amb la creació de codis per tractar les dades descarregades. Posteriorment després dels preprocessats s'afegirà la informació georeferenciada a QGIS on utilitzarà R per realitzar una anàlisi de les dades descarregades. A més, estudis geoestadístics com histogrames de freqüència i histogrames de freqüència relativa. En addició s'utilitzarà QGIS per a realitzar interpolacions dels punts de les estacions amb el mètode IDW per a la creació de mapes de les concentracions dels nivells de NO_2 a la Comunitat Valenciana simbolitzat segons els rangs implementats per la Directiva 2008/50.

Els programes de descàrrega han demostrat la seva funcionalitat i eficàcia en la recollida de dades massiva que tindrà un paper molt important en la creació de diferents models com de predicció del NO_2 a la CV.

Les metodologies creades en aquest projecte tenen un paper important en la creació de models de predicció de la contaminació i de funcionament de l'atmosfera. Aquest projecte suposa un punt de partida per a futurs projectes on l'objectiu sigui la cura de la salut de la població, el control sobre les reduccions de les emissions de contaminants i la conscienciació i la importància que tots hauríem de donar per cuidar l'entorn que ens envolta.

RESUMEN

El objetivo del proyecto es la creación de metodologías de descarga masiva de datos como estaciones meteorológicas en suelo e imágenes satelitales Sentinel 5P de distintos productos, aplicándose un tratamiento que rija orden y calidad entre los datos. Para ello se usará Python para la creación de los códigos de descarga mediante la implementación de diversas plataformas como puede ser Google Earth Engine junto a la creación de códigos para tratar los datos descargados. Posteriormente tras los preprocesados se añadirá la información georreferenciada en QGIS donde utilizará R para realizar un análisis de los datos descargados. Además, estudios geoestadísticos como histogramas de frecuencia e histogramas de frecuencia relativa. En adición se utilizará QGIS para realizar interpolaciones de los puntos de las estaciones con el método IDW para la creación de mapas de las concentraciones de los niveles de NO_2 en la Comunidad Valenciana simbolizado según los rangos implementados por la Directiva 2008/50.

Los programas de descarga han demostrado su funcionalidad y eficacia en la recogida de datos masiva que tomará un papel muy importante en la creación de distintos modelos como de predicción del NO_2 en la CV.

Las metodologías creadas en este proyecto juegan un papel importante en la creación de modelos de predicción de la contaminación, así como de funcionamiento de la atmósfera. Este proyecto supone un punto de partida para futuros proyectos donde el objetivo sea el cuidado de la salud de la población, control sobre las reducciones de las emisiones de contaminantes y la concienciación e importancia que todos deberíamos dar para cuidar el entorno que nos rodea.

SUMMARY

The objective of the project is the creation of massive data download methodologies such as ground weather stations and Sentinel 5P satellite images of different products, applying a treatment that governs order and quality among the data. Python will be used for the creation of download codes through the implementation of various platforms such as Google Earth Engine along with the creation of codes to treat the downloaded data. Subsequently, after the preprocessing, the georeferenced information will be added in QGIS where R will be used to perform an analysis of the downloaded data. In addition, geostatistical studies such as frequency histograms and relative frequency histograms. In addition, QGIS will be used to perform interpolations of the points of the stations with the IDW method for the creation of maps of the concentrations of NO_2 levels in the Valencian Community symbolized according to the ranges implemented by Directive 2008/50.

The download programs have demonstrated their functionality and efficiency in the collection of massive data that will take a very important role in the creation of different models for the prediction of NO_2 in the VC.

The methodologies created in this project play an important role in the creation of pollution prediction models as well as atmospheric performance models. This project is a starting point for future projects where the objective is to care for the health of the population, control over the reductions of pollutant emissions and the awareness and importance that we should all give to take care of the environment around us.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE

1. INTRODUCCIÓN

La industria, las edificaciones y el transporte son responsables de las emisiones contaminantes en las grandes ciudades, los principales contaminantes atmosféricos son: monóxido de carbono (CO), óxidos de nitrógeno (NO, NO_2, NO_X), dióxido de azufre (SO_2), ozono (O_3) y material particulado en suspensión (PM2.5, PM10).

1. Boina de contaminantes - Google

Las emisiones contaminantes se concentran sobre las ciudades formando boinas de contaminación (figura 1), la exposición de la población a niveles altos de los contaminantes atmosféricos puede causar graves impactos a la salud de los ciudadanos como infecciones respiratorias, derrames cerebrales, enfermedades dermatológicas, alergias, alteraciones inmunológicas, enfermedades cardíacas e incluso cáncer pulmonar. En una entrada del blog de GreenPeace [1] afirman, respaldado por datos del Instituto de Salud Global [2] (figura 2), que el 6-7% de las muertes naturales son causadas por contaminantes atmosféricos como el dióxido de nitrógeno (NO_2): "Así de brutales son los datos publicados hoy por el Instituto de Salud Global que cuantifican la mortalidad vinculada al dióxido de nitrógeno." "Pero esa boina de partículas en suspensión esconde otros gases invisibles, como el NO_2, que afecta a los seres vivos y que se traduce en miles de muertes prematuras. En las ciudades del top 10 de este ranking, liderado por Madrid, se estima que un 6-7% de las muertes naturales tienen su origen en la exposición al dióxido de nitrógeno."

Un estudio realizado por la Universidad de Harvard [3] afirma que la polución del aire ha causado la muerte de 8,7 millones de personas en 2018: "Air pollution caused by the burning of fossil fuels such as coal and oil was responsible for 8.7m deaths globally in 2018, a staggering one in five of all people who died that year, new research has found.".

MORTALITY RANKING	CITY	COUNTRY	NO_2 [ANNUAL MEAN]	AVOIDABLE DEATHS [2021 WHO LEVELS]	AVOIDABLE DEATHS [2005 WHO LEVELS]	AVOIDABLE DEATHS [LOWEST LEVELS]
1	MADRID (METROPOLITAN AREA)	SPAIN	39.2	1,966	206	2,380

2.Datos NO_2 y mortalidad - Instituto de Salud Global

Como hemos dicho anteriormente, la contaminación atmosférica genera un gran riesgo frente a la salud de las personas y es necesario un monitoreo de los niveles de contaminantes. En la región de la Comunidad Valenciana disponemos de la Red Valenciana de Vigilancia y Control de la Contaminación Atmosférica (RVVCCA) que registra distintos tipos de contaminantes. Sin embargo, antes de poder hablar de los distintos contaminantes que registran, hay que saber la composición del medio donde obtienen los datos, la atmósfera.

La atmósfera es una capa de gases retenida por la gravedad del planeta alcanzando los 1000 km de altura donde la mayor concentración de masa es el 75% en los primeros 11 km y más de la mitad de la masa en los primeros 6 km. Su principal función es la protección de la radiación ultravioleta proveniente del sol, mantener temperatura cíclica y protección contra los meteoritos para que la vida prospere en el planeta. La composición de la atmósfera la podemos observar en la siguiente figura:

COMPONENTES	FÓRMULA QUÍMICA	VOLUMEN % (AIRE SECO)
Nitrógeno	N_2	78,08
Oxígeno	O_2	20,95
Argón	Ar	0,93
Dióxodo de Carbono	CO_2	350 ppmv
Neón	Ne	18,2 ppmv
Helio	He	5,24 ppmv
Metano	CH_4	2 ppmv
Criptón	Kr	1,1 ppmv
Hidrógeno	H_2	0,5 ppmv
Oxido Nitroso	N_2O	0,3 ppmv
Xenón	Xe	0,08 ppmv
Monóxido de Carbóno	CO	0,05 - 0,2 ppmv
Ozono	O_3	0,02 - 0,03 ppmv

3.Composición atmósfera - Google

Aparte de mediciones de estaciones en suelo de la RVVCCA como hemos visto anteriormente existen otro tipo de mediciones, mediciones climatológicas como AEMET, IVIA y mediciones satelitales. Sentinel 5P es un programa de Copernicus de monitoreo y vigilancia de la atmósfera la que destacan su alta resolución temporal, es decir, el tiempo que tarda el satélite en pasar por el mismo punto sobre la Tierra. Las características de Sentinel 5P como se muestra en la tabla 1 son: resolución espacial que es el tamaño del píxel de la imagen correspondiente al terreno, la resolución temporal que es el tiempo que tarda el satélite en pasar por un mismo punto sobre la Tierra.

1.Tabla Características Sentinel 5P - Propio

SATÉLITE	RESOLUCIÓN ESPACIAL	RESOLUCIÓN TEMPORAL	RESOLUCIÓN ESPECTRAL
Sentinel 5P	7x3,5 km	1 día	Difiere según el producto

Sentinel 5P ofrece varios productos como aparece en la siguiente tabla (tabla 2):

2.Tabla Productos Sentinel 5P - Propio

SATÉLITE	PRODUCTOS
Sentinel 5P	Dióxido de nitrógeno (NO_2)
	Dióxido de azufre (SO_2)
	Metano (CH_4)
	Ozono (O_3)
	Formaldehído (HCHO)
	Monóxido de carbono (CO)
	Aerosoles

Continuando con el asunto, para la monitorización de contaminantes atmosféricos es necesario un gran volumen de datos, tanto de los mismos contaminantes como de datos meteorológicos que son un factor decisivo en cualquier proyecto que los involucre debido a la gran variabilidad que provocan los factores climatológicos como lluvia, viento, etc, en los contaminantes atmosféricos.

Este proyecto tiene el objetivo de crear una metodología para obtener ese gran volumen de datos mediante programación, incluyendo diversas plataformas y lenguajes diferentes, para que de una manera sencilla se pueda obtener una gran cantidad de datos que de otra forma sería un proceso tedioso. Además de la metodología de descarga, se ha invertido una gran cantidad de tiempo en estudiar los diferentes datos para que tengan un orden, una calidad y formato adecuado. Un ejemplo claro de lo comentado son las imágenes satelitales Sentinel 5P lanzado en octubre de 2017 y en uso desde junio del 2018 integrando Google Earth Engine, una plataforma geomática innovadora ideal para extensas cantidades de información en el que se ha tenido que aprender a trabajar, para controlar sus parámetros de calidad como pueden ser el QA (Quality assurance), Processing Status y Product Quality que son de gran importancia a la hora de tratar las imágenes. Estos parámetros indican la calidad de estos y pasarlos por alto introduciría errores en la metodología que los empleara.

Una de las particularidades del proyecto respecto a otros similares es el uso de software GIS para la visualización, como QGIS, y el análisis geoestadístico, R en QGIS, de los datos descargados. El análisis geoestadístico de los niveles de contaminación de dióxido de nitrógeno (NO_2) en la región de la Comunidad Valenciana ha sido realizado en los periodos febrero-junio de 2020 y 2021, donde se podrá visualizar mapas creados mediante interpolación de los datos, y de manera estadística los cambios de los niveles de NO_2 durante la cuarentena y el año siguiente en el mismo rango de fechas.

Para concluir, una de las metodologías de descarga realizadas es más compleja que las anteriores debido a que permite descargar automáticamente datos que solo están disponibles unas horas al día, por lo que debe permanecer en funcionamiento las 24 horas del día. Son datos meteorológicos horarios diarios proporcionados por AEMET de las estaciones en la Comunidad Valenciana, este programa dos veces al día automáticamente hace una petición al servidor de AEMET y este sube los datos proporcionados

de cada estación a un almacenamiento online llamado Dropbox donde si por alguna razón la descarga falla se genera un informe donde viene el tipo de problema. Un uso práctico del programa puede ser en el uso de modelos de predicción, ya que ofrece información meteorológica horaria diaria con el que se podrían detectar cambios de temperatura abruptos que derivan en inversiones térmicas, que son un factor importante relacionado con los contaminantes atmosféricos.

2. OBJETIVOS

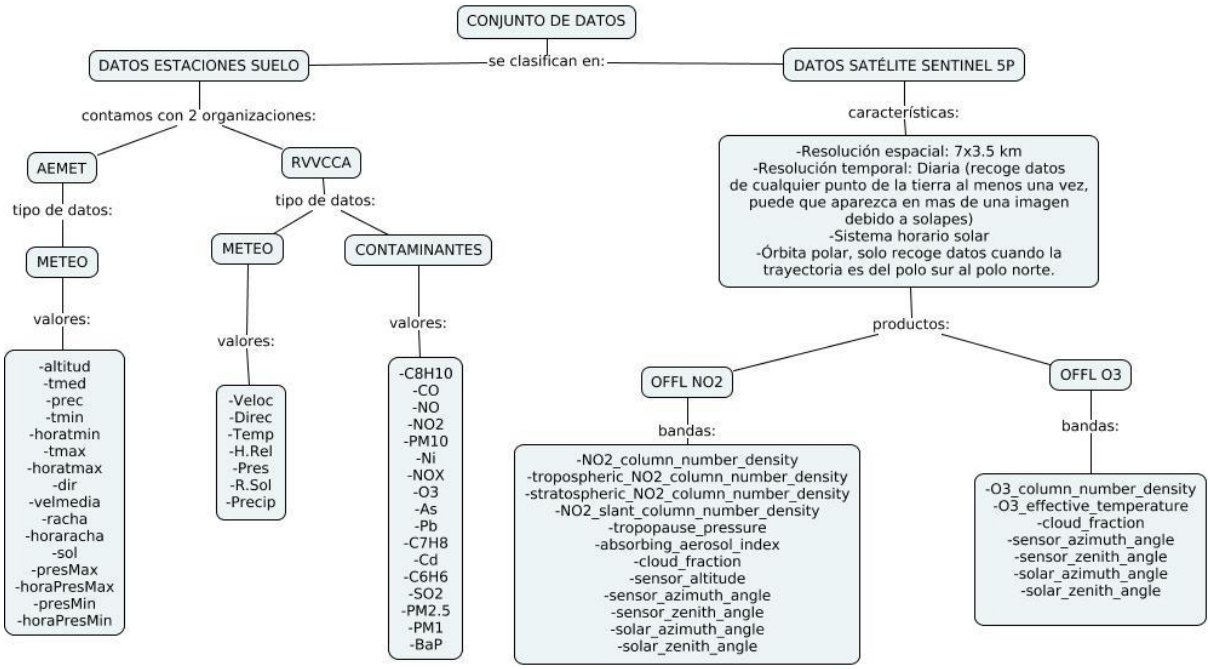
La finalidad principal del proyecto es obtener métodos de descarga masiva, mediante programación, de datos proporcionados por las estaciones meteorológicas y de contaminación en suelo e imágenes satelitales Sentinel 5P. Con los datos derivados se propone diseñar una metodología para su tratamiento que permita mantener un orden, una calidad y un formato entre ellos.

Otro objetivo complementario es realizar análisis y comparaciones geoestadísticas de los datos obtenidos. Como objetivo aplicado se propone la creación de cartografía de los niveles de dióxido de nitrógeno (NO_2) de la Comunidad Valenciana. A estos objetivos específicos se suma el reto de trabajar con un equipo formado por estudiantes del Máster en Ingeniería Geodésica, Geomática, Topografía y Cartografía y otros alumnos del grado, con el fin de desarrollar metodologías que permitan el estudio de la contaminación a diferentes escalas.

El objetivo último es la consecución del título de Graduado en el grado de Ingeniería Geodésica, Geomática, Topografía y Cartografía.

3. DATOS

En el siguiente organigrama podemos visualizar los datos utilizados y sus características (figura 4):



4.Organigrama Datos - Propio

3.1 Datos estaciones AEMET

La Agencia Estatal de Meteorología (AEMET) es una agencia estatal que proporciona servicios y datos

meteorológicos públicos para la población. Cuenta con un sistema de difusión y reutilización de los datos llamado AEMET OpenData, utilizando una clave (API KEY) que consigues al registrarte en la página web donde podrás utilizarla para descargar los datos manualmente o automatizar la recogida de datos mediante código.

Los datos obtenidos de AEMET son archivos CSV de datos meteorológicos diarios de 1994 a 2022 y datos meteorológicos en formato CSV horarios diarios mediante un programa de descarga automática que lleva recogiendo datos desde febrero hasta la actualidad.

3.2 Datos estaciones RVVCCA

La Red Valenciana de Vigilancia y Control de Contaminación Atmosférica (RVVCCA) es una red automática de control de calidad del aire en la región de la Comunidad Valenciana que proporciona mediciones de contaminantes y datos meteorológicos.

Los datos obtenidos de RVVCCA son archivos CSV de datos meteorológicos y archivos CSV de mediciones de contaminantes diarios de 1994 hasta 2022.

Estacion	FECHA (DD/MM/YYYY)	SO2 (µg/m³)	CO (mg/m³)	O3 (µg/m³)	NO2 (µg/m³)	NO (µg/m³)	PM2.5 (µg/m³)	PM10 (µg/m³)	NOx (µg/m³)	Ni (ng/m³N)	Cd (ng/m³N)	As (ng/m³N)	Pb (µg/m³)	BaP (ng/m³N)
46102002-Quart de Poblet	01/01/1994	14	0,4	67	26	12			44					
46220003-Port de Sagunt	01/01/1994												0	
46131002-Gandia	01/01/1995		0,9	81	13	8			25				0,11	
46250030-Pista de Silla	01/01/1995	15	0,8	42	24	13			44					
46102002-Quart de Poblet	01/01/1995	14	0,7	54									0,12	
46220003-Port de Sagunt	01/01/1995	6		63	15	6			25				0	
46131002-Gandia	01/01/1996		0,8	50	12	27			54				0,14	
46250030-Pista de Silla	01/01/1996		1	38	51	37			108					
46102002-Quart de Poblet	01/01/1996	8	0,6	48	30	11			47				0,15	
46220003-Port de Sagunt	01/01/1996	4	1,2	41	32	26			71				0	
46131002-Gandia	01/01/1997	3	1,2	31	26	10			42				0,1	
46250030-Pista de Silla	01/01/1997	7	1,6	18	72	77			190					
46102002-Quart de Poblet	01/01/1997	4	0,9	29	41	17			67				0,2	
46220003-Port de Sagunt	01/01/1997	3	1	35	20	18			47					
46131002-Gandia	01/01/1998	5	0,7	31	22	7			33				0,01	

5.Ejemplo CSV RVVCCA - Propio

3.3 Imágenes Sentinel 5P

Sentinel 5P es un programa de Copernicus de monitoreo y vigilancia de la atmósfera que proporciona imágenes satelitales de diferentes productos (tabla 2). Estas imágenes están almacenadas en Copernicus HUB, el portal de datos abiertos de la Agencia Espacial Europea (ESA). Este portal también cuenta con un api key como AEMET, sin embargo, no se ha utilizado para realizar la metodología de descarga. Se ha utilizado Google Earth Engine (GEE), una herramienta de Google con el que se puede acceder y operar con imágenes satelitales mediante código, para descargar las imágenes.

GEE accede a las imágenes de Copernicus HUB pero añade un procesado más, es decir si las imágenes de Copernicus HUB son de nivel 2 en GEE son de nivel 3. El nivel 3 consiste en eliminar aquellos píxeles que su quality assurance (QA, valor de calidad estándar) sea menor a un umbral establecido para cada producto. Las imágenes obtenidas de GEE son formato GeoTIFF multibanda de 2018 hasta 2021, imágenes diarias exceptuando cuando el Product Quality, un parámetro que indica si la imagen está degradada o no, indique degradación en la imagen y no sirva para operar con ella. Las características de las imágenes de Sentinel 5P son las mostradas en la tabla 1.

Sentinel 5P ofrece información de distintos contaminantes en forma de producto (tabla 2). Un producto es un conjunto de dataset de imágenes de alta resolución. El instrumento de medición es Tropomi, TROPOspheric Monitoring Instrument, un espectrómetro capaz de detectar radiación visible, ultravioleta, infrarrojo cercano y longitud de onda corta con lo que monitorea gases en la atmósfera.

Se han descargado imágenes desde la salida de los productos Sentinel 5P el 28 de junio de 2018 hasta 2021 de los productos NO_2 y O_3. El número total de imágenes descargadas de los distintos productos se puede observar en la siguiente tabla:

3.Tabla Número Imágenes Descargadas - Propio

Producto	Año 2018	Año 2019	Año 2020	Año 2021	Total Imágenes
Imágenes NO_2	253	502	504	500	1759
Imágenes O_3	249	502	504	500	1755
Total Imágenes	502	1004	1008	1000	3514

3.3.1 Producto NO_2

Tras un estudio del producto se ha concluido que las bandas seleccionadas son "tropospheric_NO2_column_number_density" y "tropopause_preassure" que son las asociadas a la troposfera. La troposfera es la capa de la atmósfera en contacto con el suelo, por ende, con la que la población está en contacto y respira. Para el monitoreo del impacto del NO_2 en la salud de las personas estas bandas serán indispensables.

Name	Units	Min	Max	Description
NO2_column_number_density	mol/m^2	-0.00051*	0.0192*	Total vertical column of NO_2 (ratio of the slant column density of NO_2 and the total air mass factor).
tropospheric_NO2_column_number_density	mol/m^2	-0.0005375*	0.0192044*	tropospheric vertical column of NO_2
stratospheric_NO2_column_number_density	mol/m^2	8.6e-06*	0.000107*	stratospheric vertical column of NO_2
NO2_slant_column_number_density	mol/m^2	1.48e-05*	0.003908*	NO_2 slant column density
tropopause_pressure	Pa	6156*	37345*	topopause pressure
absorbing_aerosol_index	Dimensionless	-14.43*	10.67*	Aerosol index (at wavelengths 354/388, i.e. the OMI pair) from the AER_AI level 2 product. See Level 2 Algorithms - Aerosol Index.
cloud_fraction	fraction	0*	1*	Effective cloud fraction. See the Sentinel 5P L2 Input/Output Data Definition Spec, p.220.
sensor_altitude	m	828543*	856078*	Altitude of the satellite with respect to the geodetic sub-satellite point (WGS84).
sensor_azimuth_angle	degrees	-180*	180*	Azimuth angle of the satellite at the ground pixel location (WGS84); angle measured East-of-North.
sensor_zenith_angle	degrees	0.09*	67*	Zenith angle of the satellite at the ground pixel location (WGS84); angle measured away from the vertical.
solar_azimuth_angle	degrees	-180*	180*	Azimuth angle of the Sun at the ground pixel location (WGS84); angle measured East-of-North.
solar_zenith_angle	degrees	8*	82*	Zenith angle of the satellite at the ground pixel location (WGS84); angle measured away from the vertical.

* estimated min or max value

6.Bandas NO_2 - Google Earth Engine

3.3.2 Producto O_3

La banda "O3_column_number density" es la única banda del producto de ozono que obtiene información de mediciones de O_3 presentes en una columna de toda la atmósfera. Esta servirá para monitorear las emisiones de ozono y comprobar si existe una relación con las mediciones de NO_2.

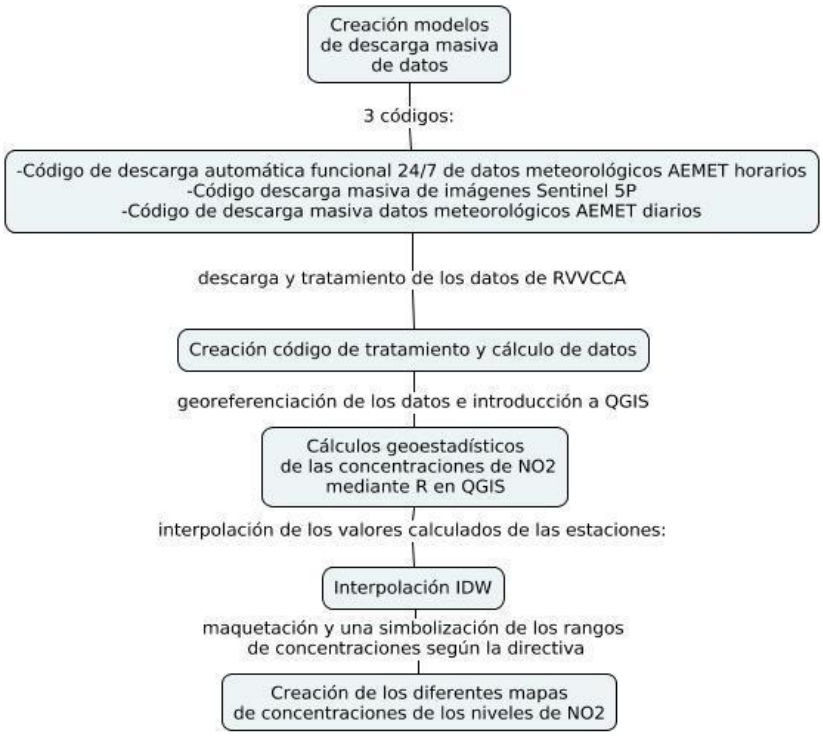
Name	Units	Min	Max	Description
O3_column_number_density	mol/m^2	0.025*	0.3048*	Total atmospheric column of O_3 between the surface and the top of atmosphere, calculated with the GODfit algorithm.
O3_effective_temperature	K	19.92*	428.11*	Ozone cross section effective temperature
cloud_fraction	fraction	0*	1*	Effective cloud fraction. See the Sentinel 5P L2 Input/Output Data Definition Spec, p.220.
sensor_azimuth_angle	degrees	-180*	180*	Azimuth angle of the satellite at the ground pixel location (WGS84); angle measured East-of-North.
sensor_zenith_angle	degrees	0.098*	66.57*	Zenith angle of the satellite at the ground pixel location (WGS84); angle measured away from the vertical.
solar_azimuth_angle	degrees	-180*	180*	Azimuth angle of the Sun at the ground pixel location (WGS84); angle measured East-of-North.
solar_zenith_angle	degrees	8*	102*	Zenith angle of the satellite at the ground pixel location (WGS84); angle measured away from the vertical.

* estimated min or max value

7.Bandas O_3 - Google Earth Engine

4. METODOLOGÍA

4.1 Organigrama



8.Organigrama metodología - Propio

4.2 Descarga automática de datos masivos meteorológicos horarios AEMET en la CV

Se ha usado python para programar el código ya que es uno de los lenguajes de programación que más ha crecido durante los años llegando a ser hoy en día uno de los más utilizados a escala global. Python además de ser versátil y eficiente tiene una gran cantidad de bibliotecas que le brinda una gran flexibilidad en el ámbito de usos que este puede ofrecer, destacando en los campos de machine learning y automatización. Se ha usado en el entorno de desarrollo integrado (IDE) Pycharm, utilizado en la programación informática, ofrece características muy versátiles a la hora de escribir código como, por ejemplo: poder ejecutar parte a parte el código para poder identificar errores y ofrece un soporte en instalación de librerías e información sobre ellas y sus funciones.

Las librerías son un conjunto de archivos realizados por otros desarrolladores que el resto de las personas aprovechan para realizar su código de una manera eficiente utilizando las diversas funciones que ofrecen estos paquetes. Por ejemplo, una librería de tratamiento de datos csv, si lo hiciera uno mismo podrían ser cientos de líneas de código mientras que usando una función de una librería sería una sola línea.

LIBRERÍAS USADAS	
Nombre	Descripción
http.client	Es una biblioteca de Python flexible, eficiente y potente para acceder a cualquier recurso de la web a través de HTTP
Json	La biblioteca json puede analizar JSON desde cadenas o archivos. La biblioteca analiza JSON en un diccionario o lista de Python. También puede convertir diccionarios o listas de Python en cadenas JSON.
Datetime	El módulo datetime proporciona clases para manipular fechas y horas.
Os	Este módulo proporciona una forma portátil de utilizar la funcionalidad dependiente del sistema operativo. Si sólo quiere leer o escribir un fichero vea open(), si quiere manipular rutas, vea el módulo os.path, y si quiere leer todas las líneas de todos los ficheros en la línea de comandos vea el módulo fileinput.
Dropbox	El SDK (software development kit) oficial de Dropbox.
Time	Este módulo proporciona varias funciones relacionadas con el tiempo.
Io	El módulo io de Python nos permite gestionar las operaciones de entrada y salida relacionadas con los archivos.
schedule	Programación de trabajos en Python, ejecuta funciones periódicamente usando una sintaxis amigable.

9.Tabla librerías código - Propio

Una de las particularidades del código es que permite descargar automáticamente datos horarios de AEMET que solo están disponibles por un límite corto de tiempo, 22-24 horas, por lo que debe permanecer activado en todo momento. El programa recorre una lista con los identificadores de las estaciones en la CV dos veces al día: una a las 8AM y otra a las 8PM, ya que los datos climatológicos horarios no registran las 24 horas anteriores completas sino 22 horas, teniendo que realizar otra descarga para completar los datos. La recogida de datos se realiza mediante peticiones junto a la apikey, servicio de opendata de AEMET, en los que se indica un condicionante de que si alguno falla se cree un informe del motivo del fallo teniendo así constancia de que día y que estación ha fallado. Los archivos recuperados de la petición son los datos y los metadatos que son almacenados en Dropbox, almacenamiento online en la nube, clasificados por el día de recogida y en número de ciclo comentado anteriormente para un mejor estructuramiento y orden de los datos.

Se ha elegido Dropbox debido a que con unos pocos clics ya tienes todo lo necesario para poder empezar a utilizarlo mediante Python, al contrario de Google Drive que hay que firmar y completar diversos apartados para poder acceder al token (es la llave de tu usuario que permite a Python conectarse con la plataforma de almacenamiento). Lo que se necesita para conectarse a una cuenta de cualquiera plataforma de almacenamiento es el token, esta se consigue mediante la creación de “aplicaciones” que no son más que una configuración de permisos con la que obtendremos el token. Ya creada la aplicación hay que configurar los diferentes permisos que le daremos como permiso para leer la información de la cuenta, escribir archivos y leerlos.

Para poder mantener “vivo” el código se necesita implementar un framework web (colección de librerías y módulos para desarrollar aplicaciones web sin necesidad de preocuparse de protocolo, gestión de hilos, etc.) mediante la librería Flask, este módulo es usado en Pinterest y LinkedIn.

Para ello se usará Replit que permite a los usuarios escribir código, crear aplicaciones y sitios web a través de un navegador sin la necesidad de instalar nada en el equipo. La librería Flask se ha utilizado en un script aparte que después se llamará en el código principal. Este script define una función que mantiene activo al código cuando es llamado (figura 10).

```python
from flask import Flask
from threading import Thread

app=Flask('')

@app.route('/')
def home():
  return "Hello, i am alive!"

def run():
  app.run(host='0.0.0.0',port=8080)

def keep_alive():
  t=Thread(target=run)
  t.start()
```

10.Keep Alive - Replit

La función home es simplemente un mensaje para comprobar que funciona correctamente, la función run es el que realiza la conexión indicando el puerto (replit lo cambia en función al tráfico) y la función keep alive es la que mantiene el código principal funcionando continuamente. En este caso los resultados son la visualización de la aplicación web con el mensaje de seguridad indicándonos que todo funciona y en la parte inferior mensajes de las conexiones y los "pings" (llamadas de una plataforma externa usada para mantenerlo conectado).

Replit ofrece codificar los tokens y api para mantener seguros las claves ya que cualquier código subido en la plataforma lo puede ver cualquiera. Además, mantiene activo el código junto a Flask durante una hora aproximadamente. El objetivo es mantenerlo encendido todo el día por lo que se usará una plataforma externa gratuita llamada UptimeRobot para hacer "pings" llamadas al código cada 5 minutos para que este no se apague además de notificar en caso de que se haya caído la página web.

4.3 Descarga masiva de datos diarios AEMET

Este código es una versión mucho más sencilla respecto al código de descarga automática de AEMET visto anteriormente, consiste en una conexión mediante el api para descargar los datos diarios de las estaciones de la Comunidad Valenciana en un rango de fechas de forma local. Recorre una lista de los identificadores y para cada estación realiza una petición que recoge los datos. Si falla la recogida de datos genera un informe del error. Cuando las iteraciones se hayan acabado se rescata la variable donde se almacenaron todos los csv y se juntan en uno solo.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
fecha	indicativo	nombre	provincia	altitud	tmed	prec	tmin	horatmin	tmax	horatmax	dir	velmedia	racha	horaracha	sol	presMax	horaPresMax	presMin	horaPresMin
01/01/1994	8019	ALICANTE-ELCHE AEROPUERTO	ALICANTE	43	16,7	0	14,2	7:50	19,2	11:00	30	4,4	20,3	10:28	1,7	1023,1	11	1019,4	2
01/01/1994	8025	ALICANTE/ALACANT	ALICANTE	81	16,6	0	13,8	3:00	19,4	12:00	33	5,6	19,2	18:15	1,5	1017,1	11	1013,4	8
01/01/1994	9563X	CASTELLFORT	CASTELLON	1220	3,9	0	0,6	7:10	7,2	0:00	30	7,8	22,5	2:10					
01/01/1994	8500A	CASTELLÓN - ALMASSORA	CASTELLON	43	13,6	0	9	20:00	18,2	14:00	27	1,9	14,2	1:34	7	1019,9	11	1013,7	1
01/01/1994	8058X	OLIVA	VALENCIA	5	15,7	0	13,3	22:05	18,1	13:00	26	5,3	18,1	5:00					
01/01/1994	8325X	POLINYÀ DE XÚQUER	VALENCIA	12	14,9	0	12,3	23:15	17,5	11:30	27	5,3	14,7	10:00					
01/01/1994	8309X	UTIEL	VALENCIA	758	8,4	0	3,5	23:00	13,3	13:50	23	4,2	14,7	13:20					
01/01/1994	8414A	VALENCIA AEROPUERTO	VALENCIA	56	15,9	0	13	23:30	18,8	15:15	27	9,4	13,9	12:00	6,3	1018,7	11	1014,7	1
01/01/1994	8416	VALÈNCIA	VALENCIA	11	16,8	0	13,8	7:30	19,8	13:45	29	5,8	13,1	6:58	5,6	1024,7	11	1020,7	1

11.Ejemplo CSV AEMET - Propio

4.4 Descarga masiva de imágenes Sentinel 5P mediante api GEE en Python

La descarga masiva de imágenes satelitales se ha realizado mediante python junto al api de Google Earth Engine en Google Colab debido a que es la forma recomendada por los desarrolladores de Google. Google Earth Engine es una plataforma de geomática en la nube basada en el lenguaje de programación JavaScript, en el que permite visualizar, analizar y calcular una gran cantidad de imágenes satelitales mediante las herramientas que proporciona junto a una potencia computacional nunca antes vista en procedimientos rudimentarios.

Como se ha comentado anteriormente, GEE proporciona herramientas de análisis junto a una gran potencia computacional de manera que la descarga de un año costaría alrededor de 30 min o una simple media de un mes costaría menos de 10 líneas de código y 5 min de tiempo. Sin embargo, si se hiciera mediante Copernicus HUB sería más tedioso ya que envuelve procesos de solicitud de imágenes que llevaría grandes tiempos de espera (sobre todo en Sentinel 2), además de que para el cálculo en ráster se tendría que añadir librerías externas lo que resultaría en un programa mucho más complejo e ineficiente. Uno de los pocos aspectos flojos de la plataforma es la descarga de datos masiva, cuando se hace la exportación de alguna imagen aparece en la pestaña de "Tasks" un botón que pone "Run" por lo que si se hace el análisis de pocas imágenes no es un inconveniente, pero cuando la suma de las imágenes llega a ser de más de tres cifras se vuelve tedioso e ineficiente. La solución es usar lo mejor de los dos mundos, es decir, usar las herramientas que proporciona GEE y la automatización de python lo que permitirá que se suban las imágenes sin la necesidad de darle al botón para subirlas.

El programa realiza la conexión api mediante la librería de GEE llamada ee, para poder utilizarla debemos autentificar nuestra cuenta de GEE mediante ee.Authenticate. Una vez ya autorizado podremos usar las diferentes herramientas y la potencia computacional en nuestro código como si estuviéramos trabajando en la misma plataforma. La configuración para descargar el producto deseado es sencilla, solo es necesario el rango de fechas, la zona de interés a descargar e indicar el producto. Se indicará el producto a descargar mediante el ID que podemos encontrar en el catálogo de GEE. Además, los datos serán filtrados por una propiedad llamada Product Quality, el que como hemos comentado anteriormente, indica si la imagen ha sido degradada provocando un arrastre de error en todo proceso que se vea envuelto.

La problemática de las imágenes de Sentinel 5P a diferencia de Sentinel 2 es que no se puede extraer una sola imagen mediante un filtro de localización (punto con coordenadas) dado que la colección no está del todo bien definida. Al filtrar por un punto con coordenadas, que debería sacar una sola imagen (o dos dependiendo si ha habido solape), te saca por pantalla las 14 imágenes del día que hace Sentinel 5P qué es la cobertura del planeta, por lo que se tenía que encontrar un método por el que solo se descargue la zona de interés. La solución que se dio para el problema de las 14 imágenes diarias fue dar con el valor máximo del píxel de la imagen recortando por la zona de interés por lo que aquellas zonas que no pertenezcan a la Comunidad Valenciana tendrán valores nulos. Un ejemplo sería si la pasada que recubre Asia del este se hace un recorte (clip) de las coordenadas de la Comunidad Valenciana los valores de los píxeles serán nulos por lo que la salida del máximo es 'None'. Este planteamiento se hace mediante el reducer.max y ee.reduceRegion donde con un condicional si el valor máximo de la imagen es diferente a 'None' realice la descarga mediante batch.Export, una herramienta creada por la comunidad para descarga masiva de imágenes, obteniendo así solo las imágenes que registren un valor máximo en la imagen que pertenece a la Comunidad Valenciana para poder exportarlas al drive.

4.5 Tratamiento de los datos meteorológicos y niveles de contaminantes

Los datos de RVVCCA no se pudo crear un código ya que no tienen un api para acceder a los opendata, se tuvo que descargar a mano estación por estación y año por año.

Los archivos descargados al ser de diferentes años y diferentes estaciones presentaban cambios en el formato que a la hora de formar un solo gran conjunto de datos dificulta la unión entre ellos, este proceso fue uno de los más tediosos al tener que ir maquetando el mismo formato a mano y no de un proceso automático. El motivo de que no se hiciera un código para maquetar los datos fue debido a la gran variabilidad de las características que había como nombres diferentes, presentaciones de los datos distintas, por ello se decidió hacerlo a mano y tener como propuesta de mejora un tratamiento automático de los datos. Sin embargo, en los últimos años se ha podido observar que han tenido en cuenta este aspecto y están manteniendo un formato entre los datos.

Ya dado el formato entre todos los archivos se juntó mediante un código que proporciona una de las academias de programación más famosas FreeCodeCamp en el que se encontró justo lo que se requería para la fusión de csv en un gran conjunto de datos.

4.6 Programa python en Jupyter Notebook de cálculo y tratamiento de los datos

Ya habiendo obtenido ese gran conjunto de datos se tenía que obtener las medias de los niveles de contaminantes en las fechas de estudio para cada estación, hacerlo de manera manual habría sido un proceso muy costoso debido al número de datos y estaciones. Sin embargo, como anteriormente ya se le había dado formato a los datos para que mantuvieran un orden entre ellos la realización de un programa para obtener las medias de los datos ha sido mucho más sencillo.

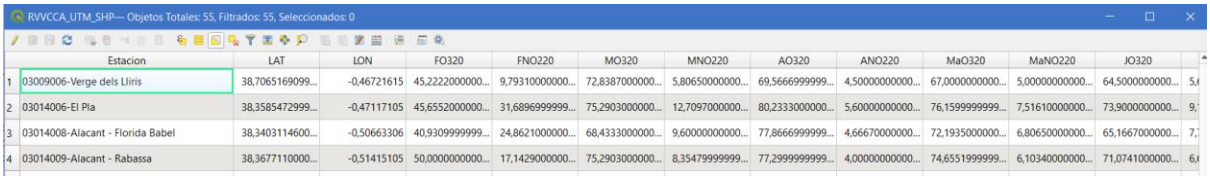
En el programa se han utilizado las librerías pandas y numpy, unas librerías muy usadas para tratamiento y cálculo de datos en formatos como csv, en el entorno de Jupyter Notebook que funciona mediante python debido a su forma fácil de visualizar por pantalla los resultados de los datos. Los parámetros a introducir son la ruta del archivo, el filtrado de las fecha y tres listas: los nombres, latitudes y longitudes de las estaciones presentes en el csv introducidos manualmente debido a un error en las recientes versiones de pandas, pero no es un problema tan grave como para que no se pueda introducir manualmente. Se recorre las listas de los nombres para filtrar según los meses de estudio con el que se obtiene las medias de esos rangos de tiempo. Posteriormente un archivo de csv para cada estación con los datos de salida al que se le indican dos filas, una con los nombres de las diferentes columnas y otras con las coordenadas de cada estación además de los cálculos de las medias correspondientes mediante numpy. Los csv se guardan en una carpeta en la que posteriormente se usa de nuevo el código proporcionado por FreeCodeCamp para poder generar un archivo csv con todas las estaciones juntos sus coordenadas y medias correspondientes. Este formato será muy útil para introducir en softwares de GIS como QGIS.

Estacion	LAT	LON	FO320	FNO220	MO320	MNO220	AO320	ANO220	MaO320	MaNO220	JO320	JNO220	FO321	FNO221	MO321	MNO221	AO321	ANO221	MaO321
03014008-Alacant - Florida Babel	38.34031146	-0.50663306	40931	24.8621	68.4333	9.6	77.8667	4.6667	72.1935	6.8065	65.1667	7.7667	56.8214	13.4643	68.8065	8.5161	83.4	4.9333	80.36
46078004-Facultats	39.50961041	-0.41796381	41931	26.8966	67.1667	11.7	64.0	5.8667	62.1935	6.5484	63.1	7.1667	53.7857	17.4643	50.9677	17.5161	64.0	13.3462	59.0357
46258001-Villar del Arzobispo	39.70801051	-0.83204256	62931	8.4483	76.7097	5.0645	75.1111	3.4815	70.0	4.2258	66.6786	4.1071	65.5926	8.2963	68.3548	4.6923	74.8148	7.4	73.2581
12040009-Ermita	39.95762675	-0.03742911	22.2069	38.7586	50.5161	12.7308	52.4	9.5667	46.8571	14.8621	45.3	14.4	47.7273	19.2143	51.36	19.7419	53.2	15.0	53.4286
12009007-Almassora - C.P.Ochando	39.94420428	-0.05739487	23.0	38.3448	54.9259	19.5667	62.6316	13.7586	57875	16.6552	57.6957	14037	46.0	20.75	52.3226	27.0333	63.2	24.8667	60.2581
46250030-Pista de Silla	39.45806013	-0.37665323	24.5862	35.6207	48.4667	16.0333	69.52	8.9667	69.1429	14.4194	64.7667	18.6333	54.8929	14.8214	57.8462	24.7778	69.2667	26.7667	63.8846
12040010-Grau	39.98353084	0.00896314	26.1034	12.0345	62.0952	11.9355	58.3	6.4643	52.3913	8.6	58.5172	7.2414	45.1786	15.25	51.6774	17.64	67.6667	11.0667	59.2581
46250048-Valencia - Moli del Sol	39.48113875	-0.40855865	27.1034	19.7241	48.7742	10.2258	65.1	7.3667	62.5161	5.7419	60.0333	6.9	55.2143	13.8636	53.5806	14.7241	67.3667	12.7	62.5484

12.CSV Medias NO_2 y O_3 - Propio

4.6 QGIS

Como se ha visto durante el grado de Ingeniería Geodésica, Geomática, Topografía y Cartografía los softwares GIS como QGIS o ARCGIS son una herramienta versátil y potente para los datos espacialmente referenciados. Por ello se ha decidido usar QGIS para realizar los distintos análisis y representaciones cartográficas de los niveles de NO_2. Para introducir los datos a QGIS ya se ha realizado previamente un tratamiento y un cálculo de los datos que se georreferenciarán gracias a las columnas de latitud y longitud, con lo que se representará en el mapa con puntos de las estaciones y esos puntos contendrán toda la información que se ha obtenido y calculado en procesos previos. Los datos de la capa de puntos son de los años 2020 y 2021 en los meses febrero-junio.


RVVCCA_UTM_SHP — Objetos Totales: 55, Filtrados: 55, Seleccionados: 0

	Estacion	LAT	LON	FO320	FNO220	MO320	MNO220	AO320	ANO220	MaO320	MaNO220	JO320	
1	03009006-Verge dels Lliris	38,7065169099...	-0,46721615	45,2222000000...	9,79310000000...	72,8387000000...	5,80650000000...	69,5666999999...	4,50000000000...	67,0000000000...	5,00000000000...	64,5000000000...	5,(
2	03014006-El Pla	38,3585472999...	-0,47117105	45,6552000000...	31,6896999999...	75,2903000000...	12,7097000000...	80,2333000000...	5,60000000000...	76,1599999999...	7,51610000000...	73,9000000000...	9,(
3	03014008-Alacant - Florida Babel	38,3403114600...	-0,50663306	40,9309999999...	24,8621000000...	68,4333000000...	9,60000000000...	77,8666999999...	4,66670000000...	72,1935000000...	6,80650000000...	65,1667000000...	7,(
4	03014009-Alacant - Rabassa	38,3677110000...	-0,51415105	50,0000000000...	17,1429000000...	75,2903000000...	8,35479999999...	77,2999999999...	4,00000000000...	74,6551999999...	6,10340000000...	71,0741000000...	6,(

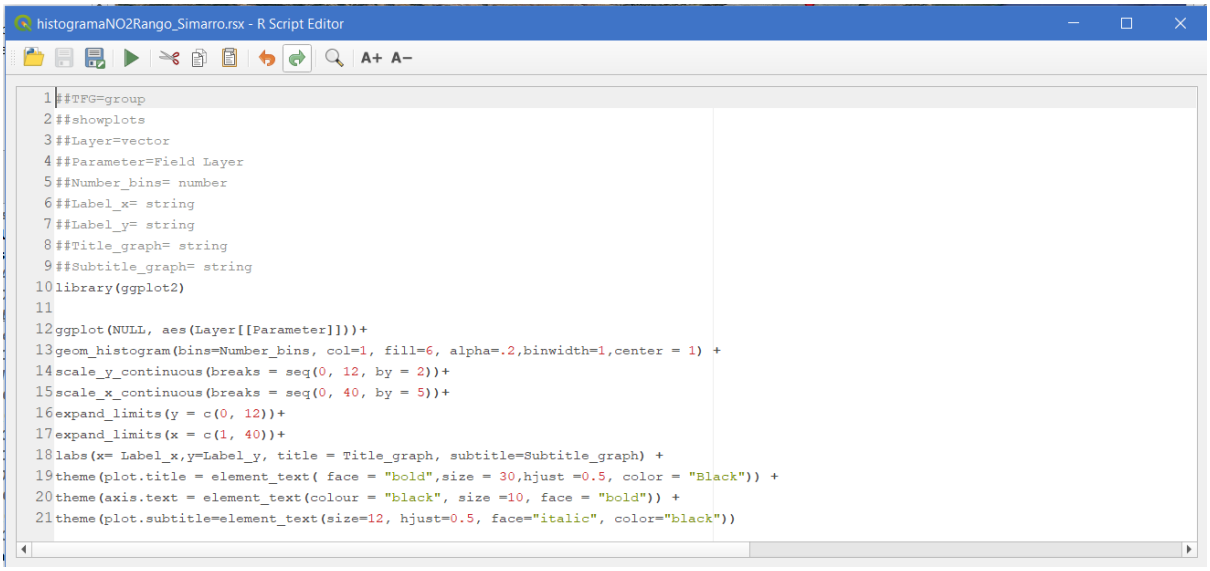
13.CSV QGIS - Propio

Se ha usado R, software de entorno y programación basado en estudios estadísticos, mediante un plugin en QGIS en el que se han creado distintos códigos para realizar dichos estudios geoestadísticos con las capas en el proyecto de QGIS. Para poder fijar un formato adecuado e igual para los diferentes histogramas se ha tenido que revisar los valores mínimos y máximos de los niveles de contaminación NO_2 con una herramienta propia del QGIS llamada estadísticas básicas, de esta forma los cambios en los histogramas se verán más contrastados.

14.Estadísticas Básicas - Propio

Los códigos creados en R para el estudio geoestadístico darán como resultado histogramas simples e histogramas de frecuencia relativa, han sido creados para poder elegir las capas de entradas desde el proyecto de QGIS y poder personalizar cómo se visualizarán en la pantalla. Para crear los histogramas se ha usado la librería ggplot2 que ofrece muchas herramientas tanto de análisis como de personalización.

Con uno de los códigos se obtiene un histograma de barras representando la frecuencia de un nivel de NO_2, las líneas que comienzan con ## son los distintos parámetros que se selecciona al iniciar el código en QGIS (figura 16) como la capa seleccionada, el campo de valores, el número de barras, nombre del eje X, nombre del eje Y, introducción de título, subtítulo y parámetro de salida.

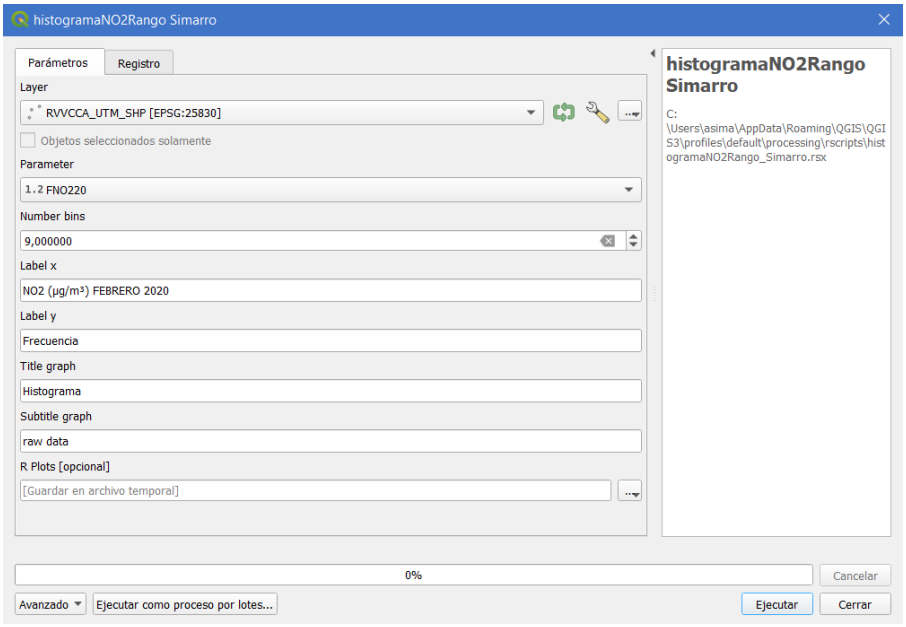


15.Código R histograma - Propio

Se ha usado la función de geom_histogram para crear el histograma de la librería ggplot2 en el que bins es el número de barras que se dividirá los datos, col que es el color, fill es el tipo de relleno de las barras, alpha la transparencia, binwidth junto a center son el rango que se regirán los datos en este caso de se hará una barra por cada valor de 1 en 1 de los niveles de NO_2 (al indicar el rango de valores para la representación no se guiará por el número de barras introducido pero se ha dejado ya que no afecta y por si en algún momento se decide usar un número predefinido de barras) y además se han indicado varios parámetros para una mejor visualización.

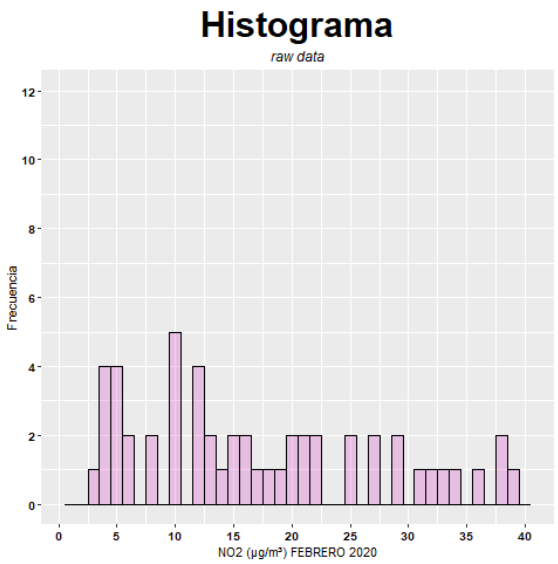
"Scale_x_continous" es un parámetro que le indica cómo tienen que seguir la visualización de los números en los diferentes ejes, es decir, de uno en uno o de dos en dos, en el caso del eje Y ha sido de 2 en dos y el eje X de 5 en 5. "Expand_limits" es un parámetro usado para expandir los existentes rangos de valores, es decir, si el máximo de marzo es 20 en un histograma este lo tomará como máximo ese valor, sin embargo, si lo añadimos se expandirá el máximo al rango de valor introducido en este caso en el eje Y a 12 y el eje X a 40.

De esta forma, aunque tengan valores máximos diferentes en los distintos meses el rango de visualización mantendrá un formato homogéneo entre ellos con el que se visualizará mayor contraste de los cambios al ponerlos juntos.



16.Menú Código Histograma - QGIS

Los resultados se pueden observar en la siguiente figura:

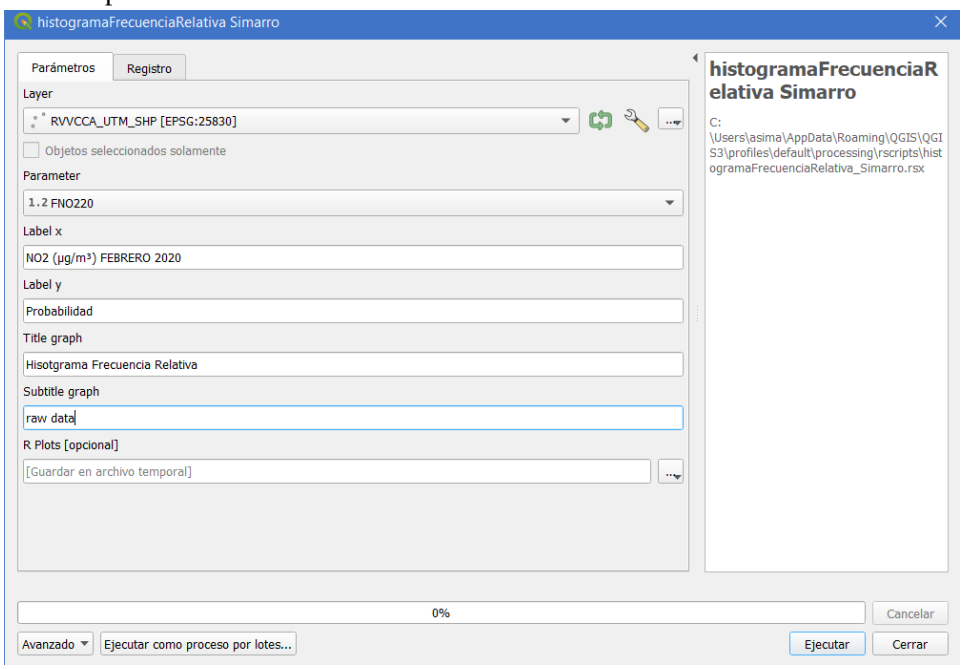


17.Ejemplo histograma - Propio

El código del histograma de frecuencia relativa es muy similar al histograma anterior, la diferencia con el otro es que en el eje Y los valores van de 0 a 1 indicando la probabilidad en porcentaje respecto al eje X de que un rango de valores sea ese respecto al total.

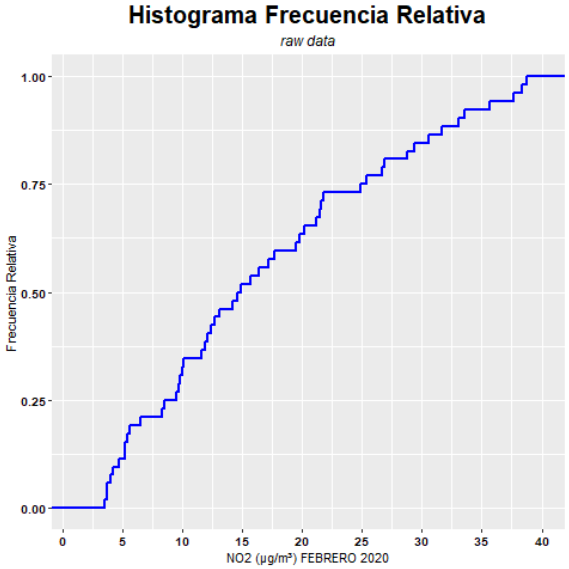
18.Código R histograma frecuencia relativa - Propio

Se ha usado la función stat_ecdf y los parámetros tanto de la función como de visualización son los mismos que en el apartado anterior.



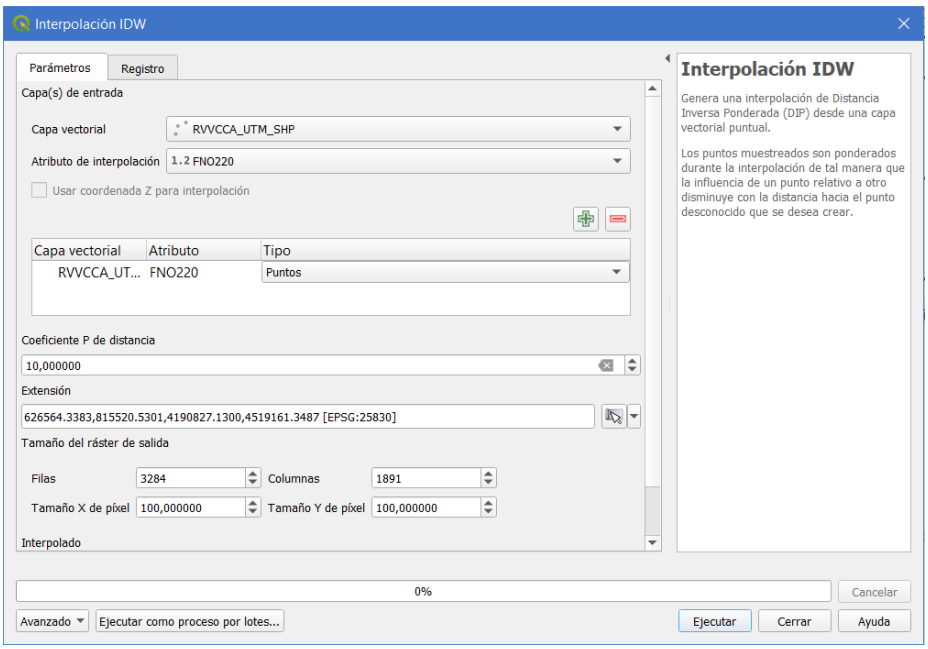
19.Menú código R Histograma Frecuencia Relativa – QGIS

Los resultados se pueden observar en la siguiente figura:



20.Ejemplo Histograma Frecuencia Relativa - Propio

La capa de las estaciones añadida a QGIS para el estudio geoestadístico es una capa vectorial, solo se sabe el valor medio de NO_2 para ese punto que es donde está la estación. Para poder saber un valor aproximado de los niveles de NO_2 donde no haya estaciones se acudirá a una interpolación de la capa formando así un ráster donde tendremos un valor de NO_2 para toda la Comunidad Valenciana a partir de las diferentes estaciones repartidas. La interpolación de Inverse Distance Weighting (IDW) o Distancia Inversa Ponderada (DIP) es una de las interpolaciones más usadas basadas en la distancia entre puntos, es decir, las zonas donde los valores son desconocidos se calculan mediante un promedio ponderando con las zonas (estaciones) que los valores son conocidos.



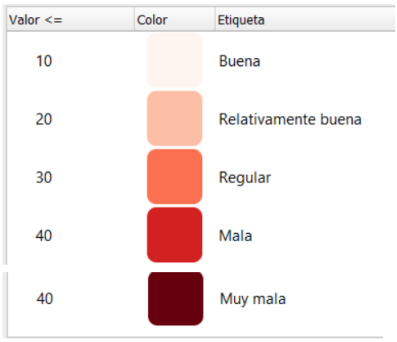
21.Menú Interpolación IDW – QGIS

Los parámetros para introducir como se puede ver en la imagen de arriba son la capa vectorial, el atributo de la interpolación a añadir, el coeficiente P de distancia que es un valor de potencia comprendido entre 0 y 99.99 a mayor valor mayor peso a los puntos cercanos a evaluar, extensión es la zona límite de la interpolación y el tamaño del ráster de salida es el tamaño que se le quiera dar al píxel.

En este proyecto como atributo de interpolación se ha usado las medias de los distintos meses con un coeficiente P de distancia de valor 10, debido a que hay estaciones más alejadas que otras y este valor es el más acertado a la realidad. Se ha comprobado que en zonas de montaña con valores P menores indican mayores niveles de NO_2 que no pueden ser posible por qué no hay poblaciones en esas zonas, pero es debido a la distancia y distribución de las estaciones, a mayor valor de P empieza a distorsionarse demasiado el ráster. La extensión es las delimitaciones territoriales de la Comunidad Valenciana y se ha elegido un tamaño de píxel de 100x100 metros.

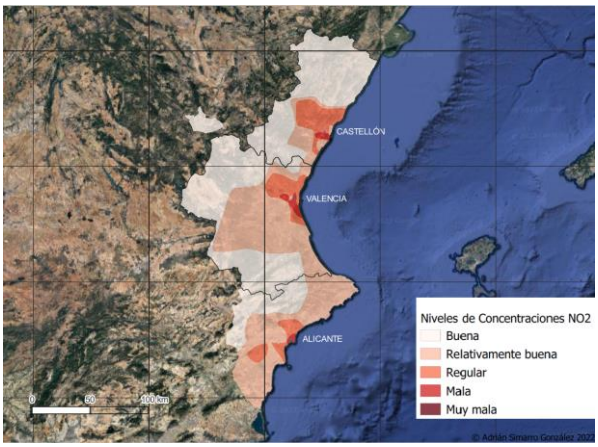
Finalmente, se crea una simbología (figura 22) para una fácil de visualización de las concentraciones de NO_2:

-Buena, valores entre 0<=10.
-Relativamente buena, valores entre 10<=20.
-Regular, valores entre 20<=30.
-Mala, valores entre 30<=40.
-Muy mala, valores 40>.



22.Simbolización - QGIS

Obteniendo así una mejor representación de las concentraciones de NO_2:



23.Simbolización de la Interpolación - Propio

5. RESULTADOS

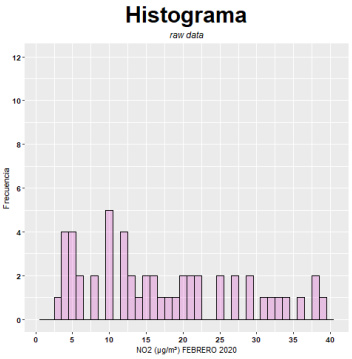
5.1 Resultados de las metodologías de descarga

El programa de descarga automática funcional las 24 horas de AEMET ha demostrado funcionar adecuadamente, además de haber sido creado mediante plataformas totalmente gratuitas ha recogido datos meteorológicos horarios diarios durante 4 meses generando un gran conjunto de datos. El programa de descarga de imágenes satelitales Sentinel 5 mediante GEE ha demostrado funcionar adecuadamente y con una rapidez asombrosa para la descarga de años de imágenes. La duración de las descargas tarda una hora aproximadamente en comprobar las 5110 imágenes anuales de cualquier producto de Sentinel 5P para solo descargar nuestra zona de interés. El programa de descarga de datos AEMET utilizando el api ha demostrado funcionar correctamente indicando los filtros de tiempo o servicio que se quiera indicar para realizar la descarga de una forma sencilla y local.

5.2 Resultados de los análisis geoestadísticos de las concentraciones NO_2

Los resultados estadísticos cuentan con dos histogramas, uno es un histograma de frecuencia (figura 24) y el otro es un histograma de frecuencia relativa (figura 25) basado en la probabilidad. A continuación, se explicará cómo interpretar cada uno de los histogramas:

El histograma de frecuencias es una relación de las concentraciones de los niveles de NO_2 y cuantas veces se repiten, en el eje X tenemos las concentraciones NO_2 de 0 a 40 (µg/m³) en rangos de 1 por 1 y en el eje Y tenemos la frecuencia, es decir, cuantas veces se repiten esas concentraciones.



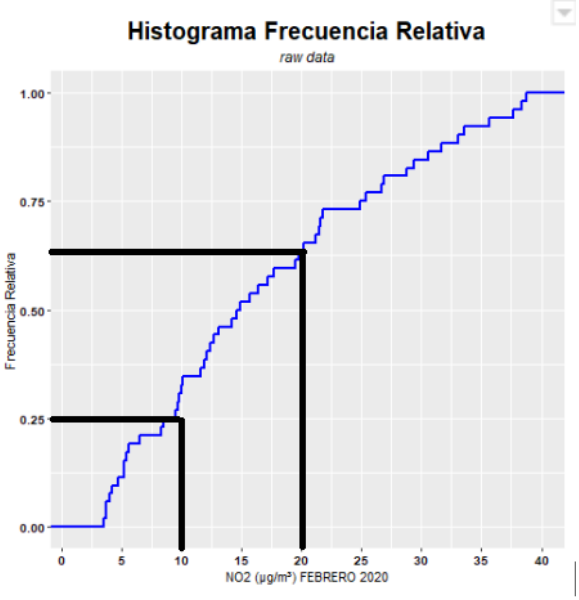
24.Histograma Simple - Propio

El histograma de frecuencia relativa muestra la probabilidad que tiene una medida de NO_2 menor o igual a la misma sobre el total.

A continuación, se explicará un ejemplo en concreto mirando el histograma de frecuencia relativa:

La probabilidad de obtener un valor menor o igual a 10 (µg/m³) representado como P(10>=x) es del 25%, es decir, P(10>=x)=0.25 (figura 25). Si se quiere saber cuál sería la probabilidad de obtener un valor que no esté comprendido entre 0 y 10 (µg/m³) simplemente tenemos que restar a la probabilidad anterior menos uno, P(10<x)=1-0.25=0.75, la probabilidad de obtener un valor mayor a 10 (µg/m³) sería del 75%.

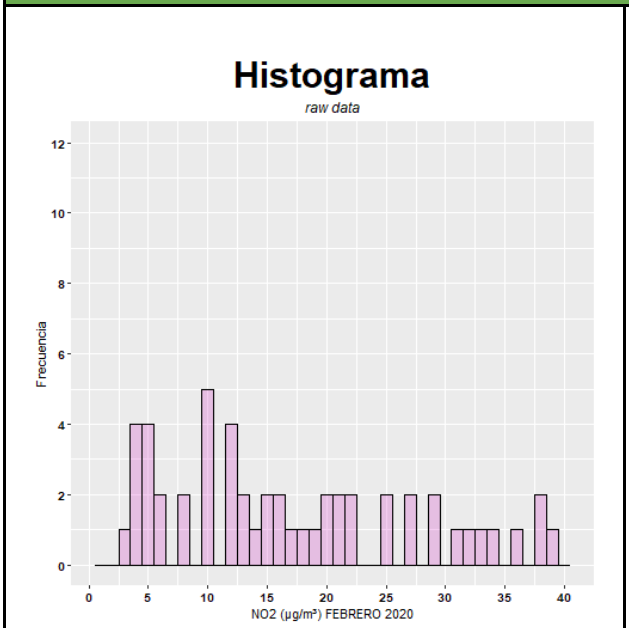
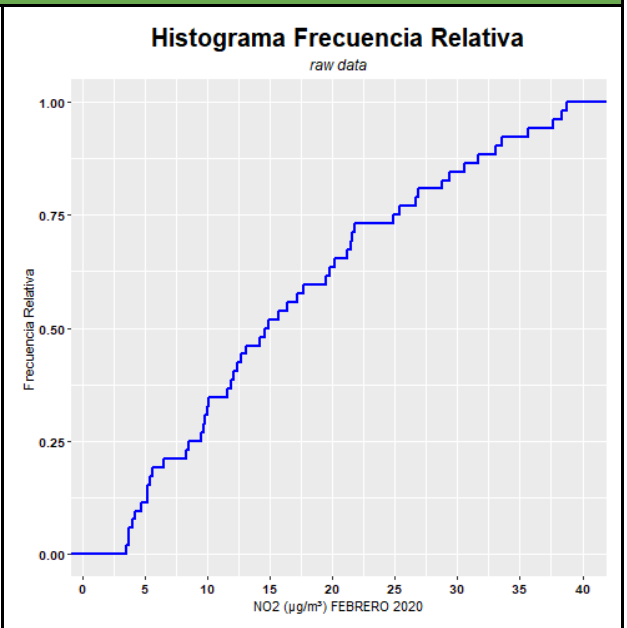
Si lo que se quiere es obtener un rango entre 10 y 20 (µg/m³), primero tendremos que obtener P(20>=x)=0.625 y restarlo a P(10>=x), es decir, P(10<=x<=20)=0.625-0.25=0.375 por lo que la probabilidad de que un valor esté comprendido entre ese rango es del 37.5%.



25.Histograma de Frecuencia Relativa - Propio

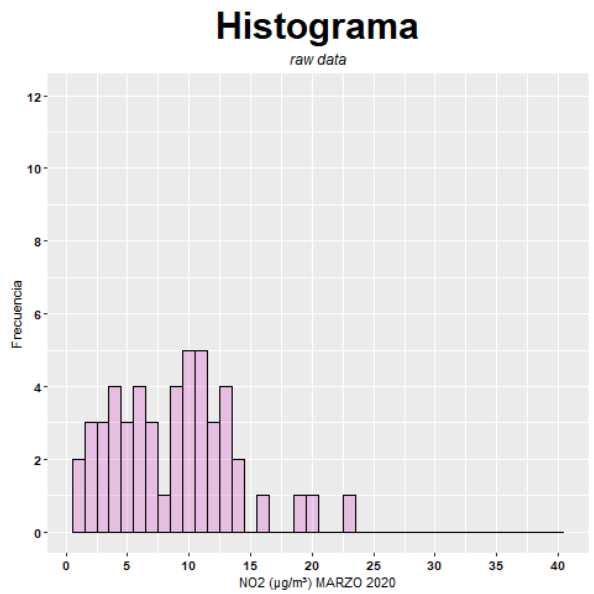
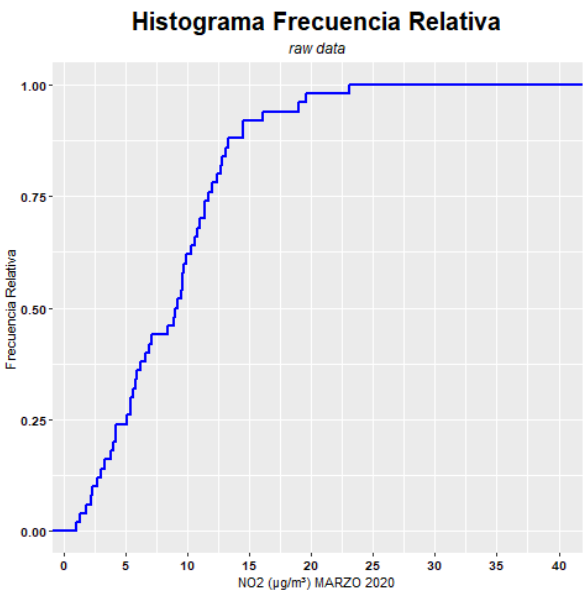
5.2.1 Histogramas año 2020

4.Tabla Febrero 2020 - Propio

FEBRERO 2020		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	18	P(10>=x)= 0.25
10<=20 Relativamente Buena	16	P(10<=x<=20)=0.63-025=0.38
20<=30 Regular	10	P(20<=x<=30)=0.85-0.63=0.22
30<=40 Mala	8	P(30<=x<=40)=1-0.85=0.15
>40 Muy Mala	0	P(40<x)=0
TOTAL	52	1

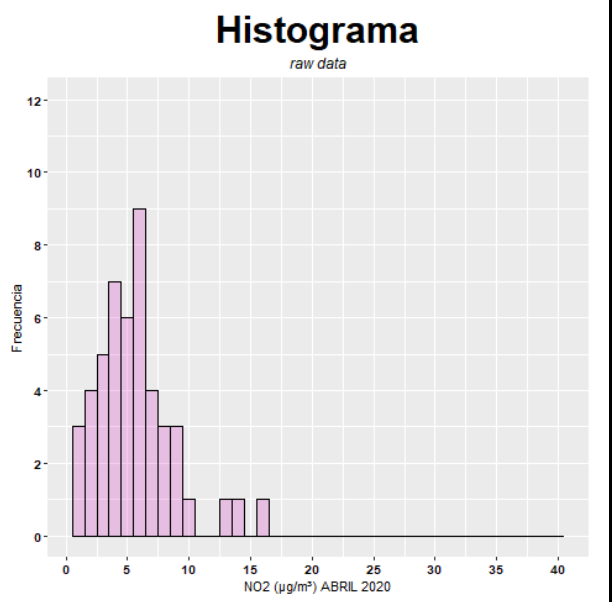
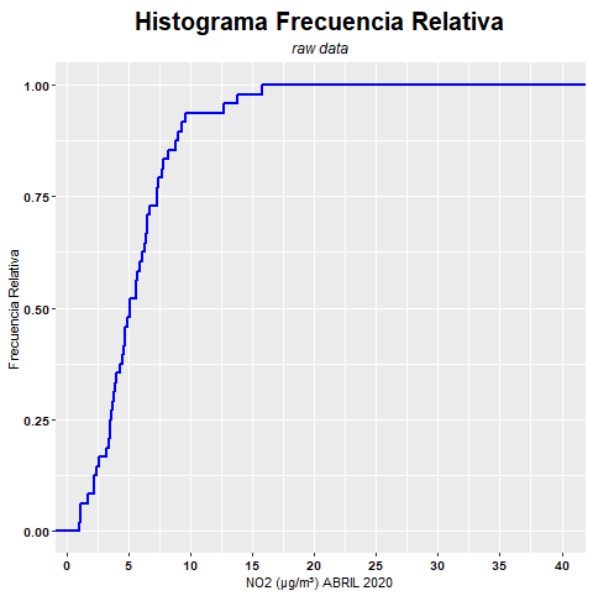
El mes de febrero de 2020, un mes anterior a la cuarentena, como se ve en la tabla predominan los valores medios comprendidos entre 10 a 30 (µg/m³) superiores a la considerada buena por la directiva con una probabilidad del 60%. Respecto a las concentraciones consideradas buenas llegan al 25 % de probabilidad. Las concentraciones comprendidas entre el 30 y 40 (µg/m³) consideradas malas tienen un 15% de probabilidad ubicadas en las capitales de las provincias como se puede ver en la tabla de las interpolaciones.

5.Tabla Marzo 2020 - Propio

MARZO 2020		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	32	P(10>=x)=0.62
10<=20 Relativamente Buena	17	P(10<=x<=20)=0.99-0.62=0.37
20<=30 Regular	1	P(20<=x<=30)=1-0.99=0.01
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	50	1

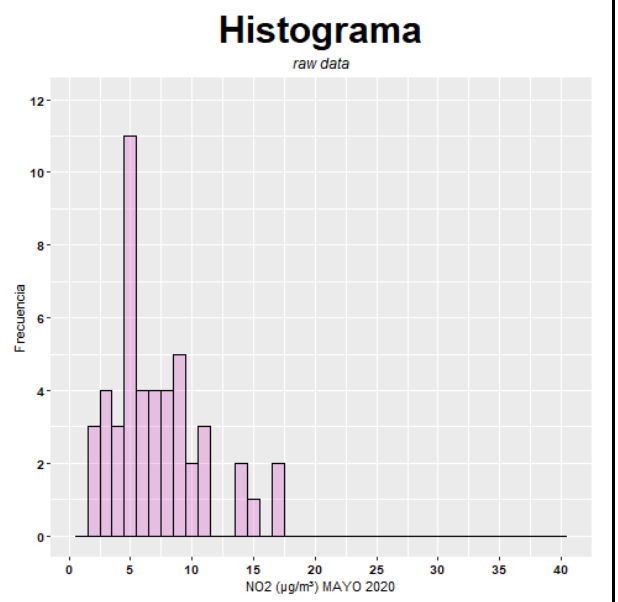
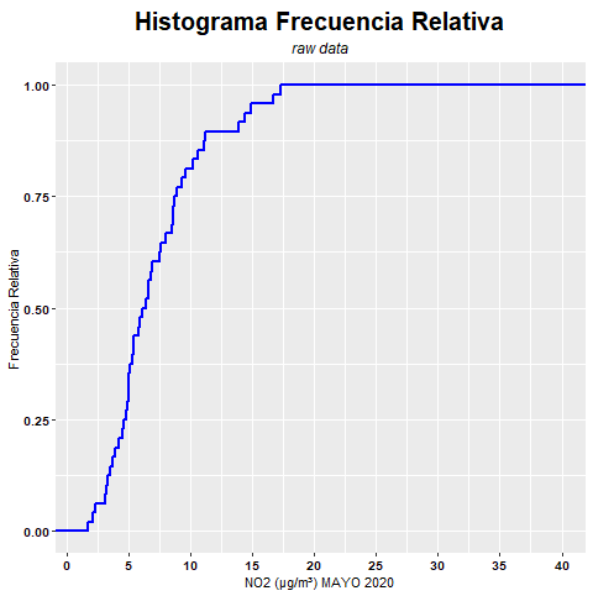
El 15 de marzo en España se introdujo una cuarentena que duraría hasta el 21 de junio, en el que con dos semanas de confinamiento se puede observar un gran aumento de las mediciones consideradas buenas llegando a casi doblar la frecuencia debido a la desaparición de mediciones superiores a 30 (µg/m³) y la disminución de la mayoría de las mediciones superiores a 20 (µg/m³). Comparado con febrero la probabilidad de las mediciones menor o iguales a 10 (µg/m³) ha pasado del 25% al 62% mientras que las mediciones relativamente buenas han permanecido prácticamente igual.

6.Tabla Abril 2020 - Propio

ABRIL 2020		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	45	P(10>=x)= 0.91
10<=20 Relativamente Buena	3	P(10<=x<=20)=1-0.91=0.09
20<=30 Regular	0	P(20<=x<=30)=0
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	48	1

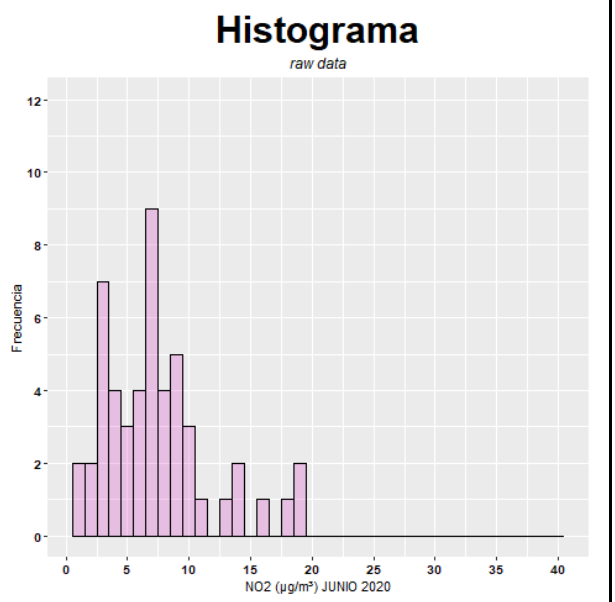
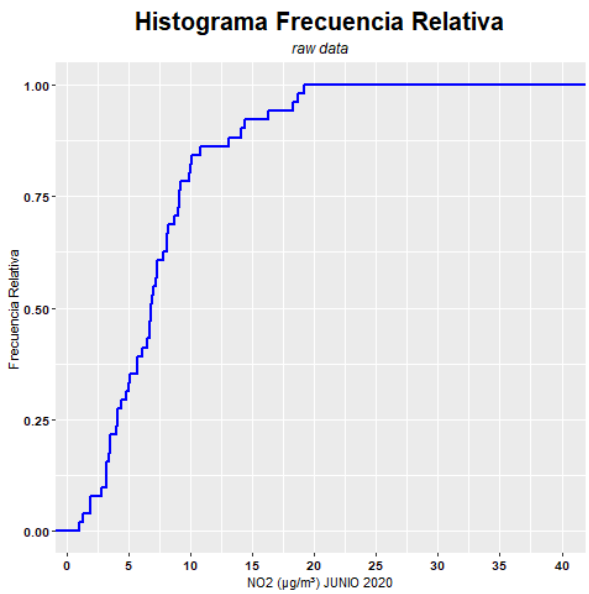
Siguiendo la evolución respecto marzo, se puede observar el continuo incremento de las mediciones buenas pasando de probabilidades del 62% al 91% haciéndose con la mayoría predominante de los datos. Las mediciones relativamente buenas descienden de un 37% de probabilidad a un 9% para sumarse a las consideradas buenas. Finalmente, los grupos de mediciones superiores a 20 (µg/m³) desaparecen completamente.

7.Tabla Mayo 2020 - Propio

MAYO 2020		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	40	P(10>=x)= 0.81
10<=20 Relativamente Buena	8	P(10<=x<=20)=1-0.81=0.19
20<=30 Regular	0	P(20<=x<=30)=0
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	48	1

En mayo los cambios son mínimos, un pequeño descenso en las mediciones buenas e incremento de las mediciones consideradas relativamente buenas. Los valores mayores a 20 (µg/m³) siguen sin registrarse durante el confinamiento.

8.Tabla Junio 2020 - Propio

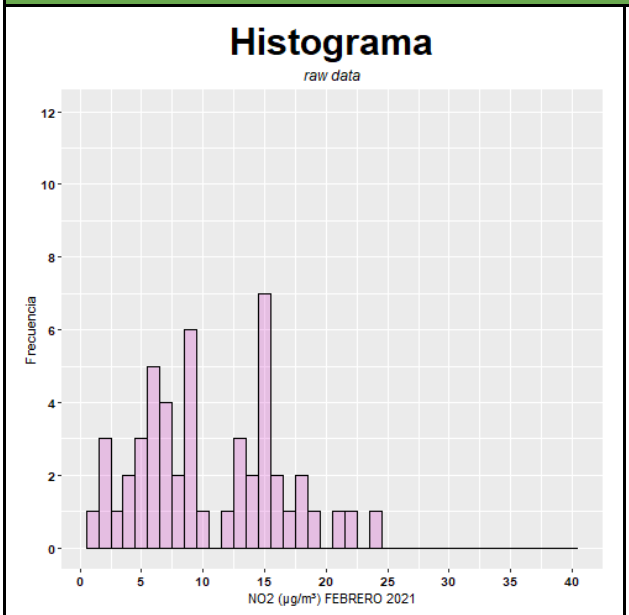
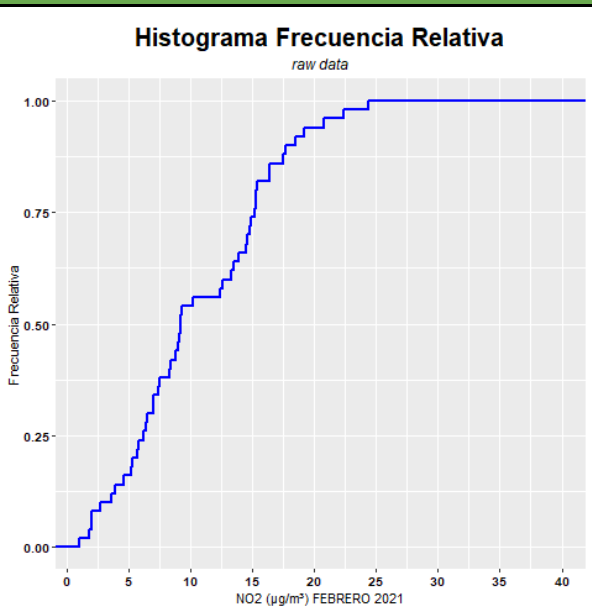
JUNIO 2020	
	

Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	43	P(10>=x)= 0.81
10<=20 Relativamente Buena	8	P(10<=x<=20)=1-0.81=0.19
20<=30 Regular	0	P(20<=x<=30)=0
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	51	1

La disponibilidad de algunas estaciones que se habían suspendido durante la cuarentena se ha reanudado, sin embargo, de mayo a junio se ha mantenido constante sin cambios significativos.

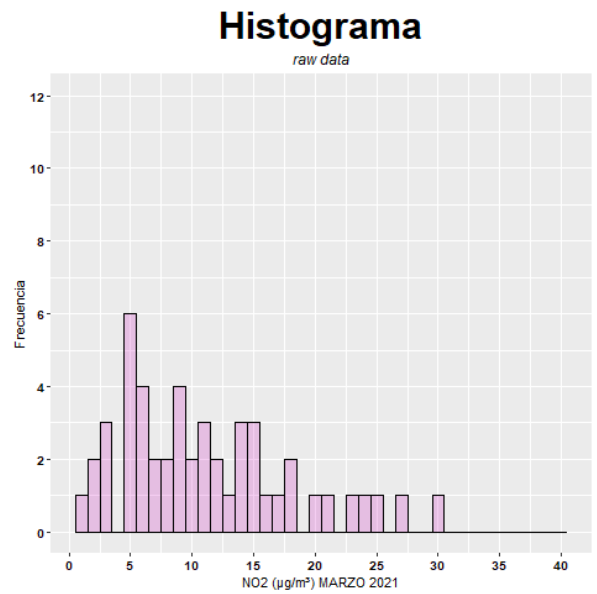
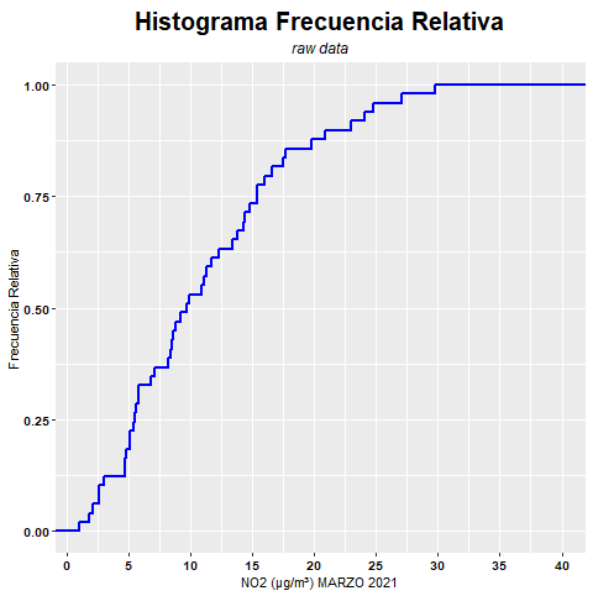
5.2.2 Histogramas año 2021

9.Tabla Febrero 2021 - Propio

<table>
<tr><th colspan="3">FEBRERO 2021</th></tr>
<tr><td colspan="2"> 
</td><td> 
</td></tr>
<tr><th>Rangos según Directiva</th><th>Frecuencia</th><th>Probabilidad</th></tr>
<tr><td><=10 Buena</td><td>28</td><td>P(10>=x)= 0.54</td></tr>
<tr><td>10<=20 Relativamente Buena</td><td>19</td><td>P(10<=x<=20)=0.94-0.54=0.40</td></tr>
<tr><td>20<=30 Regular</td><td>3</td><td>P(20<=x<=30)=1-0.94=0.06</td></tr>
<tr><td>30<=40 Mala</td><td>0</td><td>P(30<=x<=40)=0</td></tr>
<tr><td>>40 Muy Mala</td><td>0</td><td>P(40<x)=0</td></tr>
<tr><td>TOTAL</td><td>50</td><td>1</td></tr>
</table>

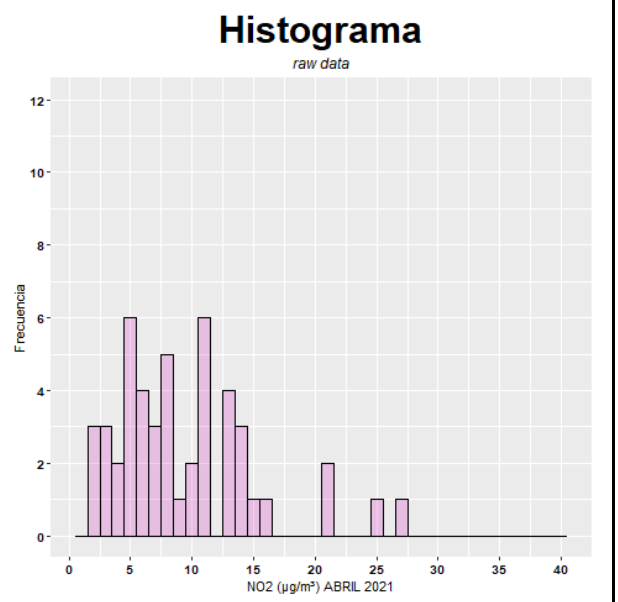
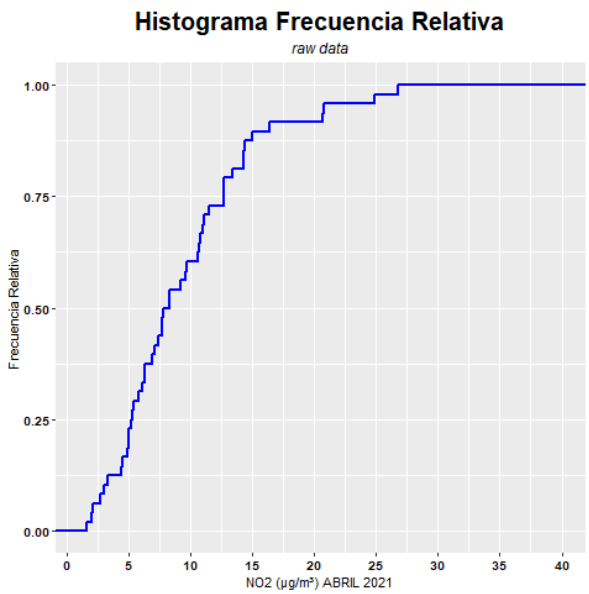
En cuanto a febrero de 2021 las mediciones buenas han aumentado respecto a febrero de 2020, han pasado de un 25% a un 54% de probabilidad cuando el 21 de junio ya acabó la cuarentena. Los valores mayores a 20 (µg/m³) han disminuido y mantenido como las primeras semanas de confinamiento.

10.Tabla Marzo 2021 - Propio

MARZO 2021		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	26	P(10>=x)= 0.52
10<=20 Relativamente Buena	17	P(10<=x<=20)=0.85-0.52=0.33
20<=30 Regular	6	P(20<=x<=30)=1-0.85=0.15
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	49	1

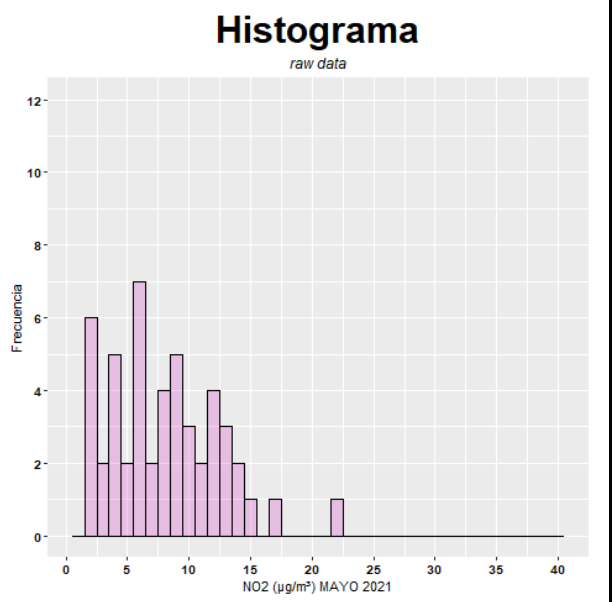
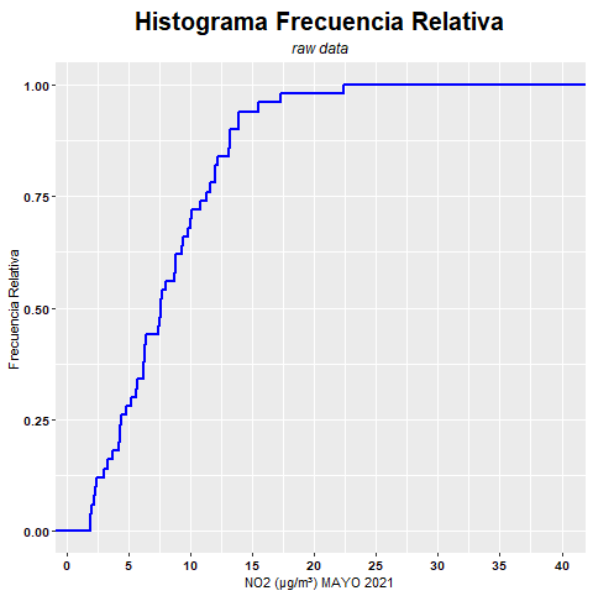
No ha habido cambios significativos respecto al mes anterior, se ha mantenido constante.

11.Tabla Abril 2021 - Propio

ABRIL 2021		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	29	P(10>=x)= 0.58
10<=20 Relativamente Buena	15	P(10<=x<=20)=0.92-0.58=0.34
20<=30 Regular	4	P(20<=x<=30)=1-0.92=0.08
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	48	1

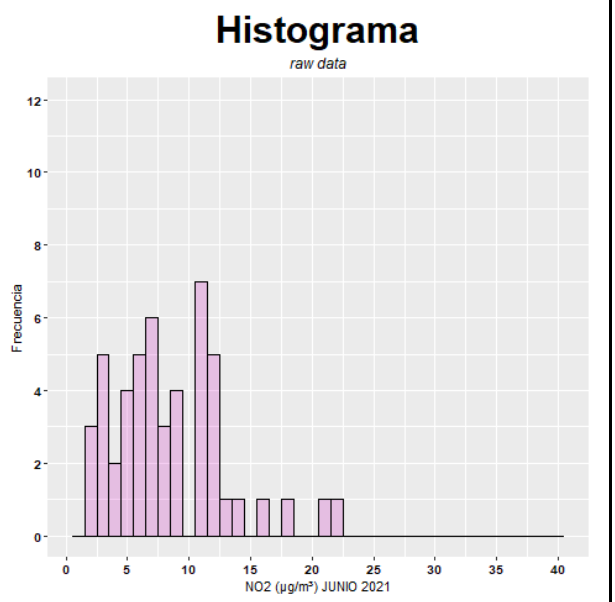
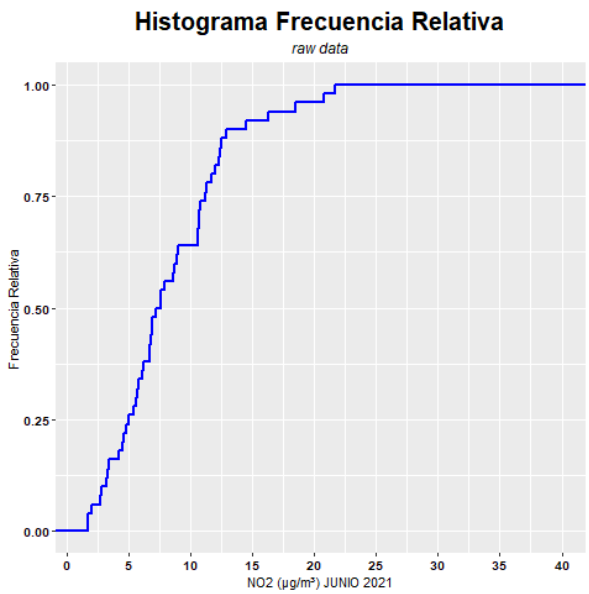
Ha habido un pequeño aumento de las mediciones consideradas buenas respecto a marzo.

12.Tabla Mayo 2021 - Propio

MAYO 2021		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	36	P(10>=x)= 0.66
10<=20 Relativamente Buena	13	P(10<=x<=20)=0.98-0.66=0.32
20<=30 Regular	1	P(20<=x<=30)=1-0.98=0.02
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	50	1

Siguen en aumento las mediciones consideradas buenas mientras que las relativas van disminuyendo y solo hay una medición considerada regular.

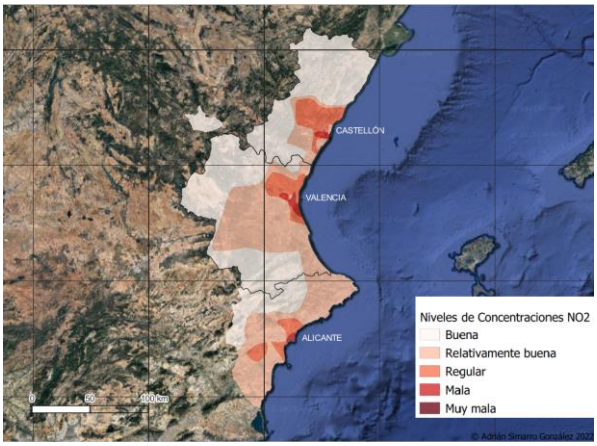
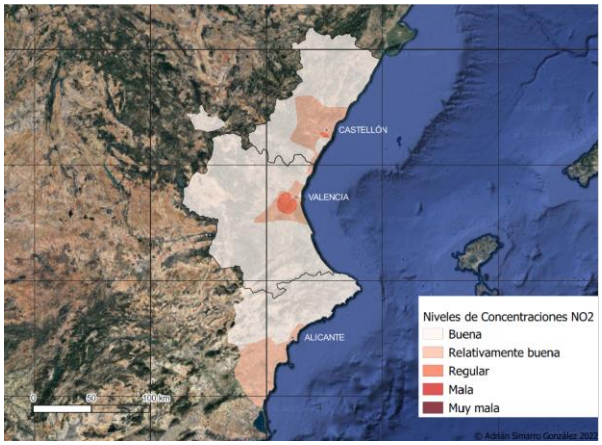
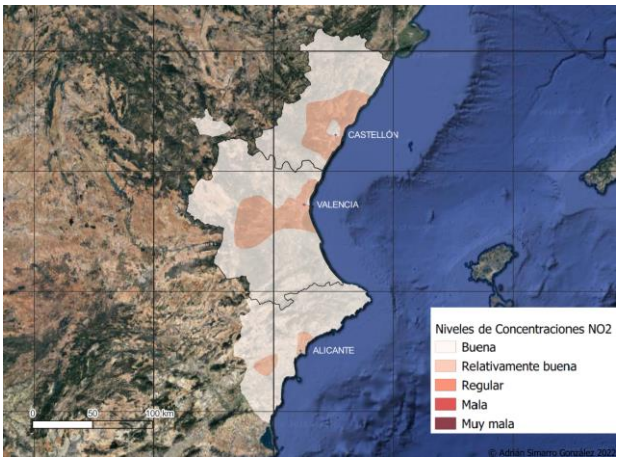
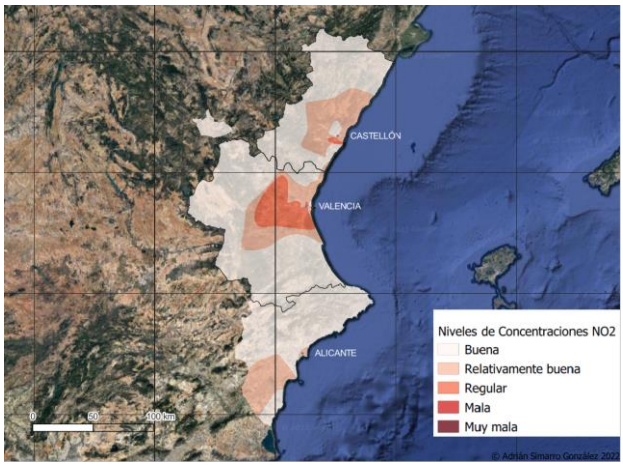
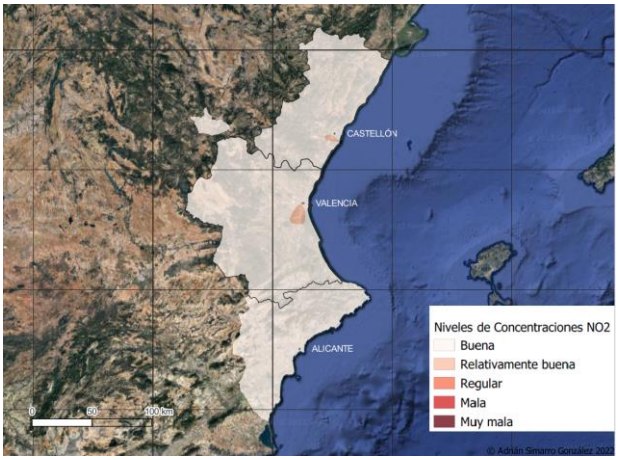
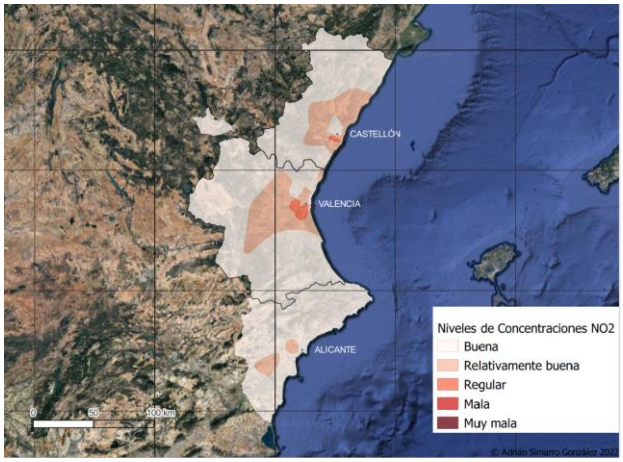
13.Tabla Junio 2021 - Propio

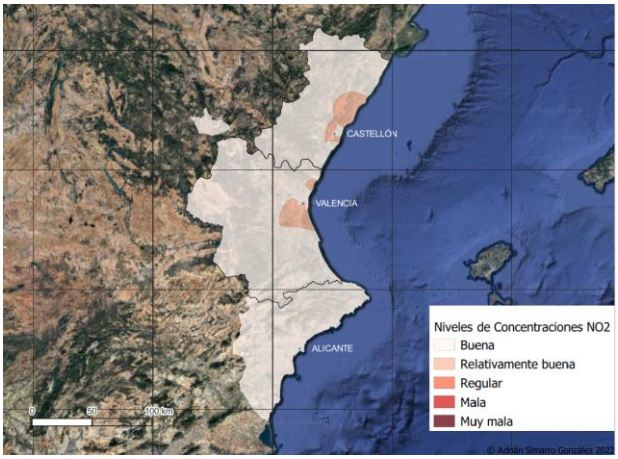
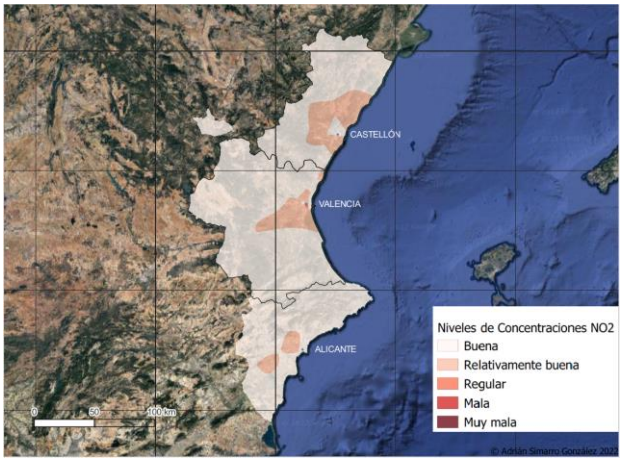
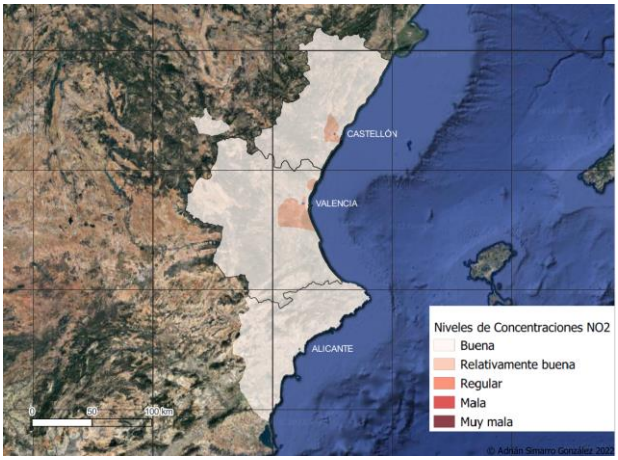
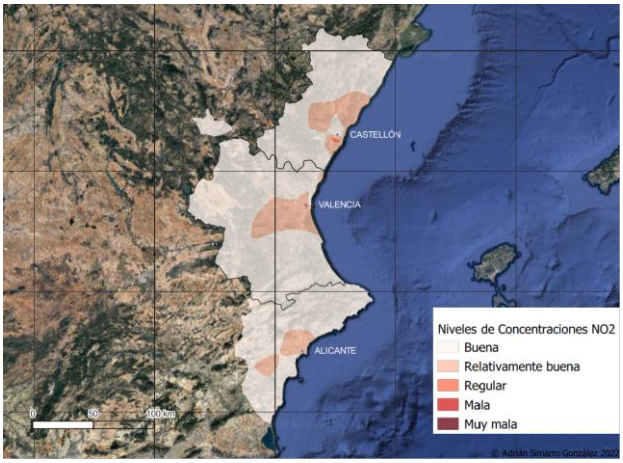
JUNIO 2021		
		
Rangos según Directiva	Frecuencia	Probabilidad
<=10 Buena	32	P(10>=x)= 0.64
10<=20 Relativamente Buena	16	P(10<=x<=20)=0.96-0.64=0.32
20<=30 Regular	2	P(20<=x<=30)=1-0.96=0.04
30<=40 Mala	0	P(30<=x<=40)=0
>40 Muy Mala	0	P(40<x)=0
TOTAL	50	1

No ha habido cambios significativos, desde febrero las mediciones buenas han pasado de tener una probabilidad del 54% al 64% con unos pequeños picos en las mediciones intermedias.

5.2 Interpolaciones de las concentraciones de NO_2 años 2020 y 2021

14.Tabla Mapas concentraciones NO_2 -Propio

MES	AÑO 2020	AÑO 2021
FEBRERO		
MARZO		
ABRIL		

MAYO		
JUNIO		

Se puede observar en los distintos mapas de las concentraciones de NO_2 en 2020 la disminución de los niveles debido a la pandemia, mientras que en 2021 los valores vuelven a subir, pero no tanto como los que presentaban en febrero de 2020, con picos pequeños de variaciones en las zonas pobladas.

6. PRESUPUESTO

Los datos de salarios del personal ingeniero geomático se han recogido del BOE Convenio de ingenieros y oficinas técnicas del año 2020.

15.Tabla Presupuesto Personal - Propio

Personal	Salario Base	Salario Bruto	Plus por Convenio	Sueldo Bruto Anual	Seguridad Social
Ing Geomático	1.291,04 €	18.074,56 €	2.349,69 €	20.424,25 €	8.169,70 €

16.Tabla Presupuesto Personal Diario - Propio

Coste Total Empresa	Coste X Día de Trabajo
28.593,95 €	129,97 €

17.Tabla Presupuesto Materiales - Propio

Material	Licencia	Uso en Proyecto
PYTHON	Gratuita	3 meses
GEE	Gratuita	4 meses
REPLIT	Gratuita	1 mes
UPTIMEROBOT	Gratuita	1 mes
GOOGLE COLAB	Gratuita	1 meses
QGIS	Gratuita	2 semanas
R	Gratuita	2 semanas
JUPYTER NOTEBOOK	Gratuita	2 días
Excel	UPV	2 semanas

18.Tabla Presupuesto Actividades y Total - Propio

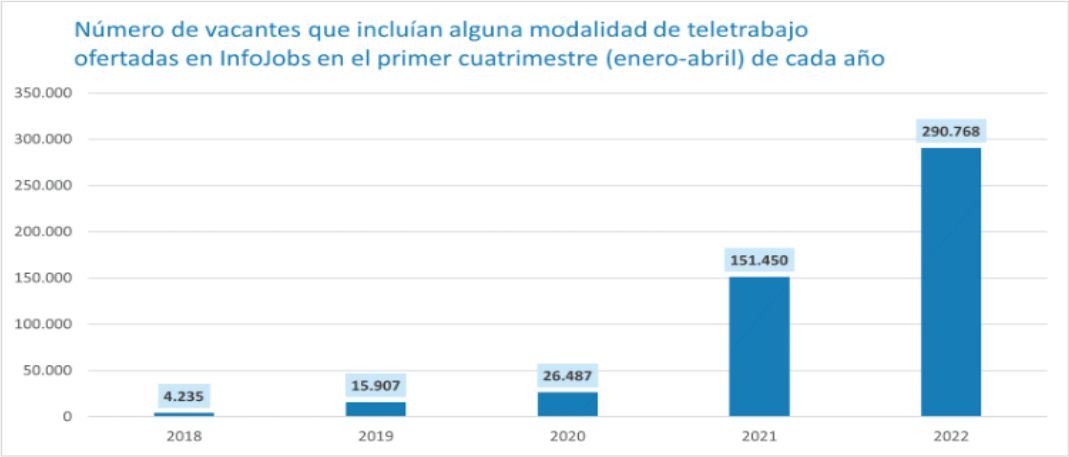
Actividades	Duración en Horas	Recursos Humanos	Coste Recursos Humanos	Medios Materiales	Coste Materiales	Total
Creación programas AEMET	160 horas	Ing Geomático	866,47 €	Python	0 €	866,47 €
				Replit	0 €	
				UptimeRobot	0 €	
Aprendizaje GEE	400 horas	Ing Geomático	2.166,17 €	GEE	0 €	2.166,17 €
Creación programa descarga imágenes Sentinel 5 y descarga de las imágenes	240 horas	Ing Geomático	1.299,7 €	GEE	0 €	1.299,7 €
				Python	0 €	
				Google Colab	0 €	
Tratamiento datos	112 horas	Ing Geomático	606,53 €	Excel	0 €	606,53 €
Programa de tratamiento y cálculo de datos	12 horas	Ing Geomático	64,99 €	Jupyter Notebook	0 €	64,99 €
				Pyhton	0 €	
Aprendizaje de R y de los análisis geoestadóisticos	56 horas	Ing Geomático	303,27 €	Internet	0 €	303,27 €
				QGIS	0 €	
				R	0 €	
Análisis geoestadísticos en R mediante QGIS	40 horas	Ing Geomático	216,62 €	R	0 €	216,62 €
				QGIS	0 €	
Generación cartografía	24 horas	Ing Geomático	129,97 €	QGIS	0 €	129,97 €
Total						5.653,72 €

7. CONCLUSIONES

De acuerdo con los objetivos planteados, se ha demostrado el funcionamiento y eficacia de los programas de descarga de datos masivos para los datos de las estaciones e imágenes satelitales Sentinel 5P.

Los análisis geoestadísticos y mapas de interpolaciones han permitido observar claramente una disminución de las concentraciones de los niveles de NO_2 en 2020. Según un artículo del National Geographic España [6] respaldado por estudios de la ESA, la disminución de la contaminación ambiental ha disminuido gracias al parón tanto industrial como movilístico: "El parón brusco de las actividades humanas tiene, paradójicamente, un gran beneficiado: el medio ambiente. El descenso de la cantidad de desplazamientos en vehículos a motor, la disminución de la producción industrial y el consumo se traduce en menos contaminación, aguas más limpias y cielos más claros. Desde China hasta Venecia, Barcelona o Madrid, estos son algunos de los efectos secundarios positivos de la crisis sanitaria.".

Respecto a los niveles de concentraciones de NO_2 en 2021, se han observado pocos cambios entre los meses observados, pero no han vuelto a esos valores altos que presentaban en febrero de 2020. Según un informe de InfoJobs [7], bolsa de empleo privada en línea, ha habido un incremento de las ofertas de teletrabajo en la pandemia que luego se multiplicaron en los años siguientes: "El teletrabajo irrumpió con fuerza durante el confinamiento estricto al comienzo de la crisis sanitaria de la covid hace dos años. Y, aunque el panorama desde entonces ha cambiado mucho, el trabajo en remoto es una realidad al alza en España: el número de vacantes registradas en el portal de empleo InfoJobs que recogían algún tipo de modalidad de teletrabajo durante los primeros cuatro meses de 2022 duplica las cifras del mismo periodo de 2021 y multiplica por 10 las de 2020.".



26.Histograma Aumento Teletrabajo - InfoJobs

Lo que explicaría por qué tras un año del confinamiento las concentraciones de NO_2 se han reducido y no han vuelto a cómo solían ser en 2021.

7.1 Propuestas de mejora

Uno de los aspectos que se podrían mejorar del proyecto es en la concepción de un programa de descarga de datos de la RVVCCA que como se ha comentado anteriormente no disponía de un api para poder acceder fácilmente mediante código. La idea tras el programa es "descargar" la página y desglosar la información para poder posteriormente almacenarla, de manera que se podría automatizar y hacer una recolección diaria automática que sería clave para modelos de predicción.

7.2 Aplicaciones

Las metodologías creadas juegan un papel muy importante en la creación de modelos de contaminantes atmosféricos como modelos de predicción, los cuales requieren un gran volumen de datos y podrán ayudar a generar modelos para cuidar el bienestar de la salud de la población, reducir las emisiones de contaminantes y concienciar para cuidar el entorno que nos rodea.

8. BIBLIOGRAFÍA

[1] La boina nos cuesta años de vida. *Greenpeace España*. Disponible en: https://es.greenpeace.org/es/noticias/boina-madrid/

[2] Rankings - ISGlobal Ranking Of Cities. *ISGlobal Ranking Of Cities*. Disponible en: https://isglobalranking.org/ranking/spain#air

[3] MILMAN, Oliver. 'Invisible killer': fossil fuels caused 8.7m deaths globally in 2018, research finds. *the Guardian* [en línea]. 9 de febrero de 2021. Disponible en: https://www.theguardian.com/environment/2021/feb/09/fossil-fuels-pollution-deaths-research

[4] Karn Vohra, Alina Vodonos, Joel Schwartz, Eloise A. Marais, Melissa P. Sulprizio, Loretta J. Mickley .*Global mortality from outdoor fine particle pollution generated by fossil fuel combustion: Results from GEOS-Chem*

[5] Atmosfera. *Programa 60+ Digital*. Disponible en: http://www7.uc.cl/sw_educ/contam/atm/atm06.htm#:~:text=En%20la%20actualidad,%20la%20atmósfera,reaccionar%20con%20ningún%20otro%20elemento.

[6] ALCALDE, Sergi. El planeta, el principal beneficiado por el coronavirus. *www.nationalgeographic.com.es* 24 de diciembre de 2020. Disponible en: https://www.nationalgeographic.com.es/ciencia/planeta-principal-beneficiado-por-coronavirus_15325

[7] *El teletrabajo registra cifras récord en España* Mayo de 2022. Disponible en: https://nosotros.infojobs.net/prensa/notas-prensa/el-teletrabajo-registra-cifras-record-en-espana

Stack Overflow - Where Developers Learn, Share, & Build Careers. *Stack Overflow*. Disponible en: https://stackoverflow.com

FREECODECAMP.ORG. *Code a Discord Bot with Python - Host for Free in the Cloud . YouTube*. 15 de diciembre de 2020. Disponible en: https://www.youtube.com/watch?v=SPTfmiYiuok

Developers. *Dropbox: Homepage*. Disponible en: https://www.dropbox.com/developers/

UptimeRobot: Free Website Monitoring Service. *UptimeRobot: Free Website Monitoring Service*. Disponible en: https://uptimerobot.com

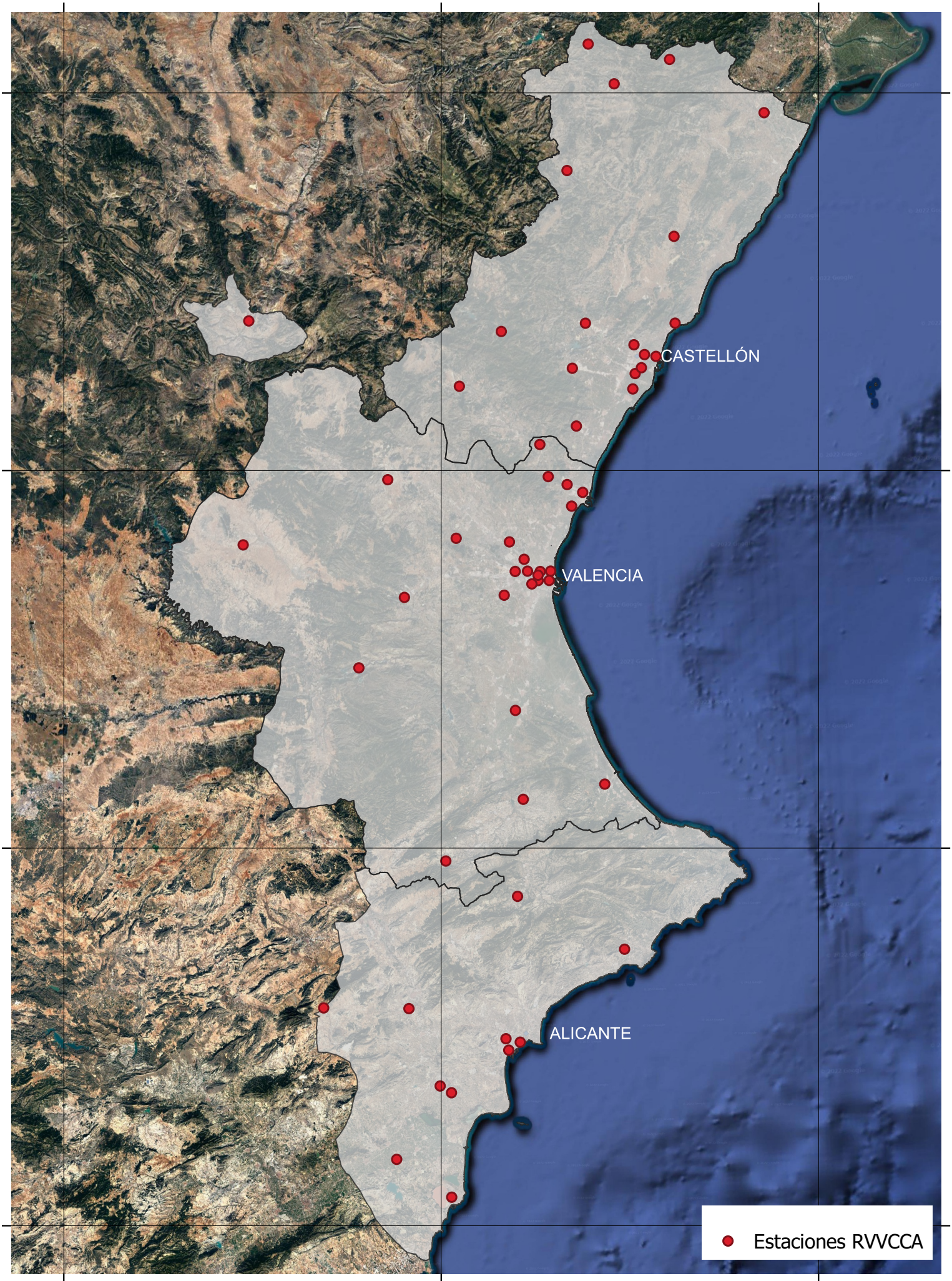
CONTRIBUTORS TO WIKIMEDIA PROJECTS. Flask (web framework) - Wikipedia. *Wikipedia, the free encyclopedia*. 3 de octubre de 2010. Disponible en: https://en.wikipedia.org/wiki/Flask_(web_framework)

The collaborative browser based IDE. *replit*. Disponible en: https://replit.com

9. CARTOGRAFÍA

9.1 Mapa estaciones RVVCCA

9.2 Mapas concentraciones NO_2 en la Comunidad Valenciana

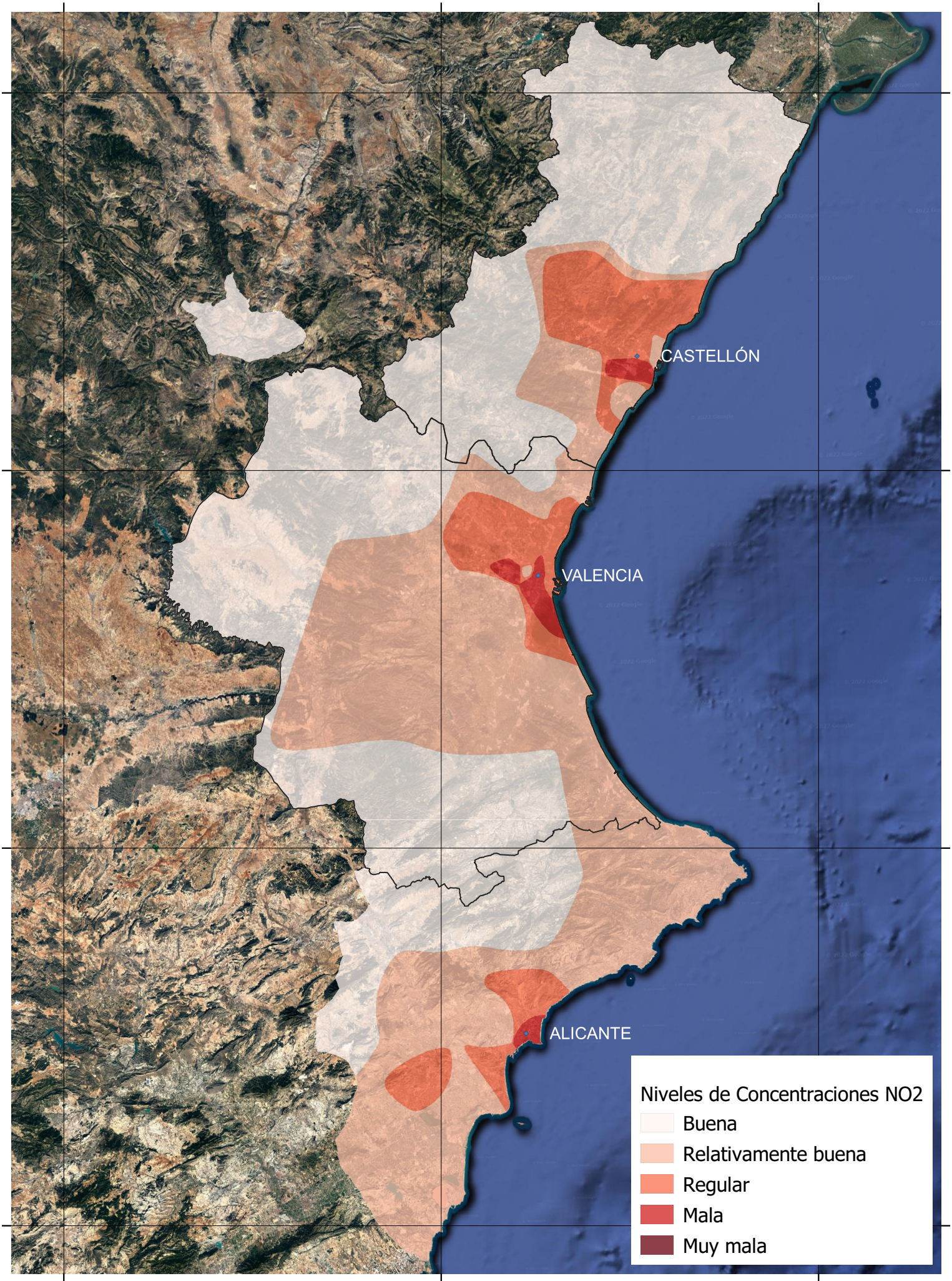


MAPA ESTACIONES DE LA RVVCCA EN LA COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

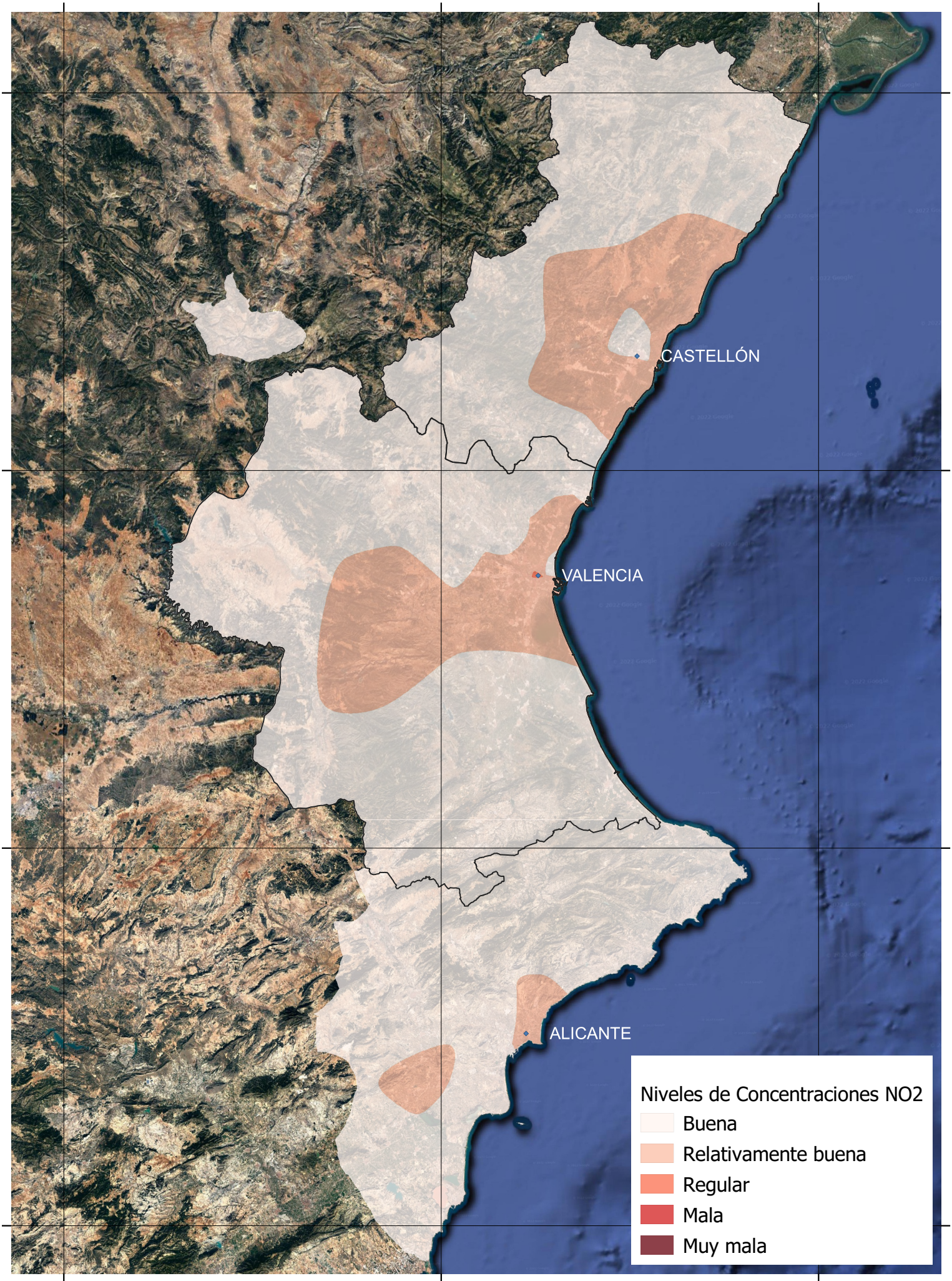


MAPA CONCENTRACIONES NO2 FEBRERO 2020 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA



MAPA CONCENTRACIONES NO2 MARZO 2020 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

MAPA CONCENTRACIONES NO2 ABRIL 2020 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

MAPA CONCENTRACIONES NO2 MAYO 2020 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

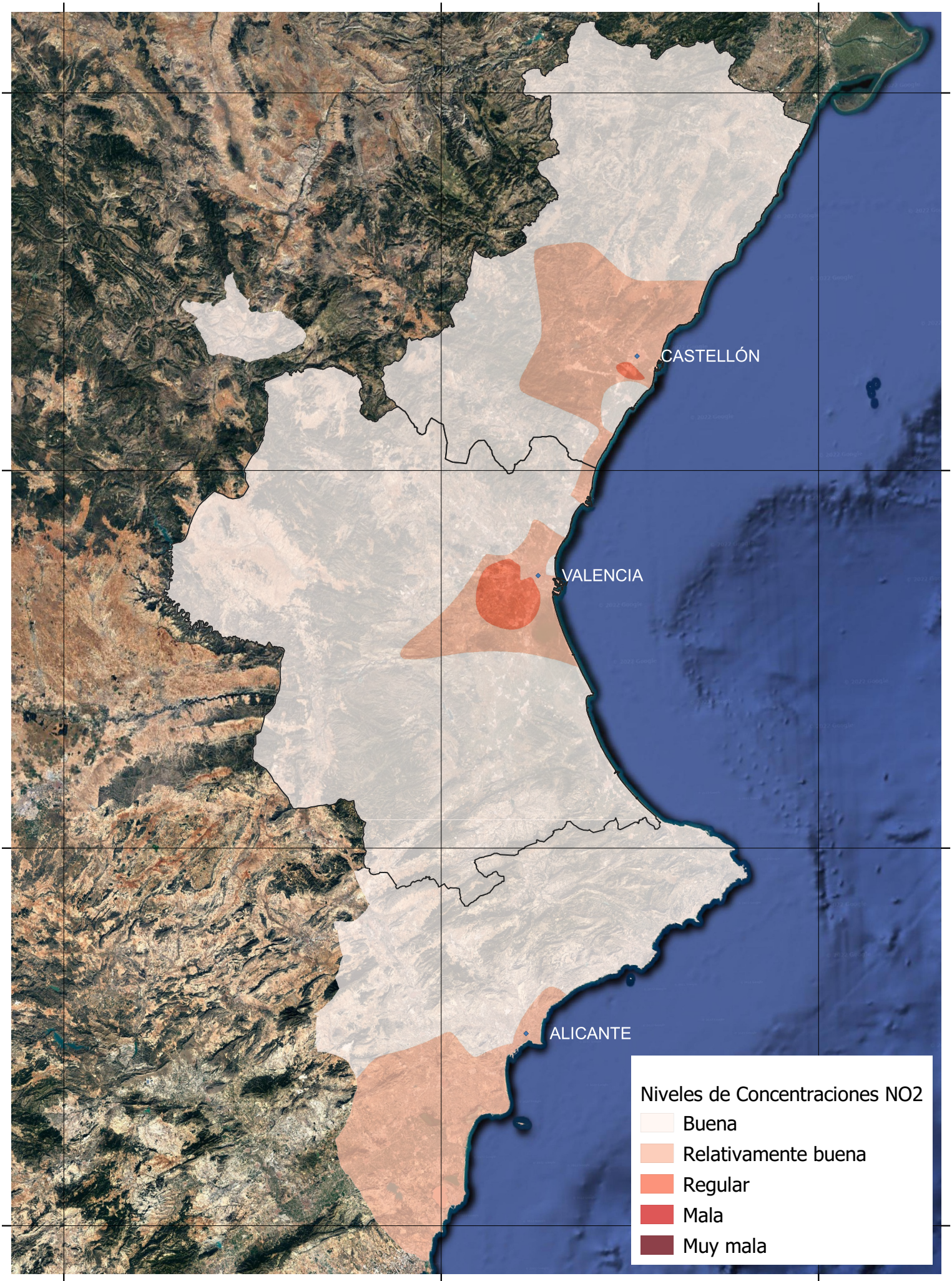
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

MAPA CONCENTRACIONES NO2 JUNIO 2020 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

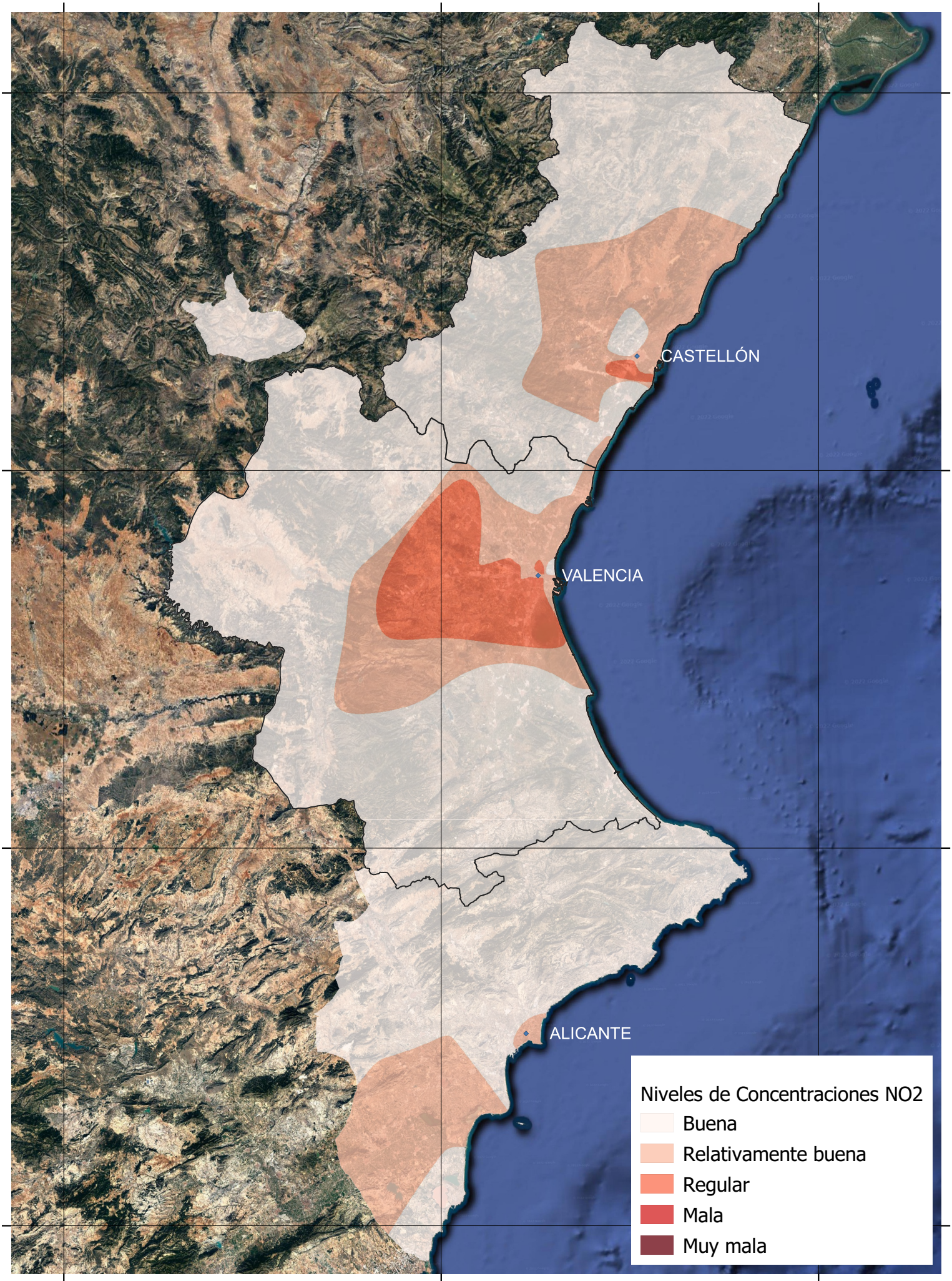


MAPA CONCENTRACIONES NO2 FEBRERO 2021 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

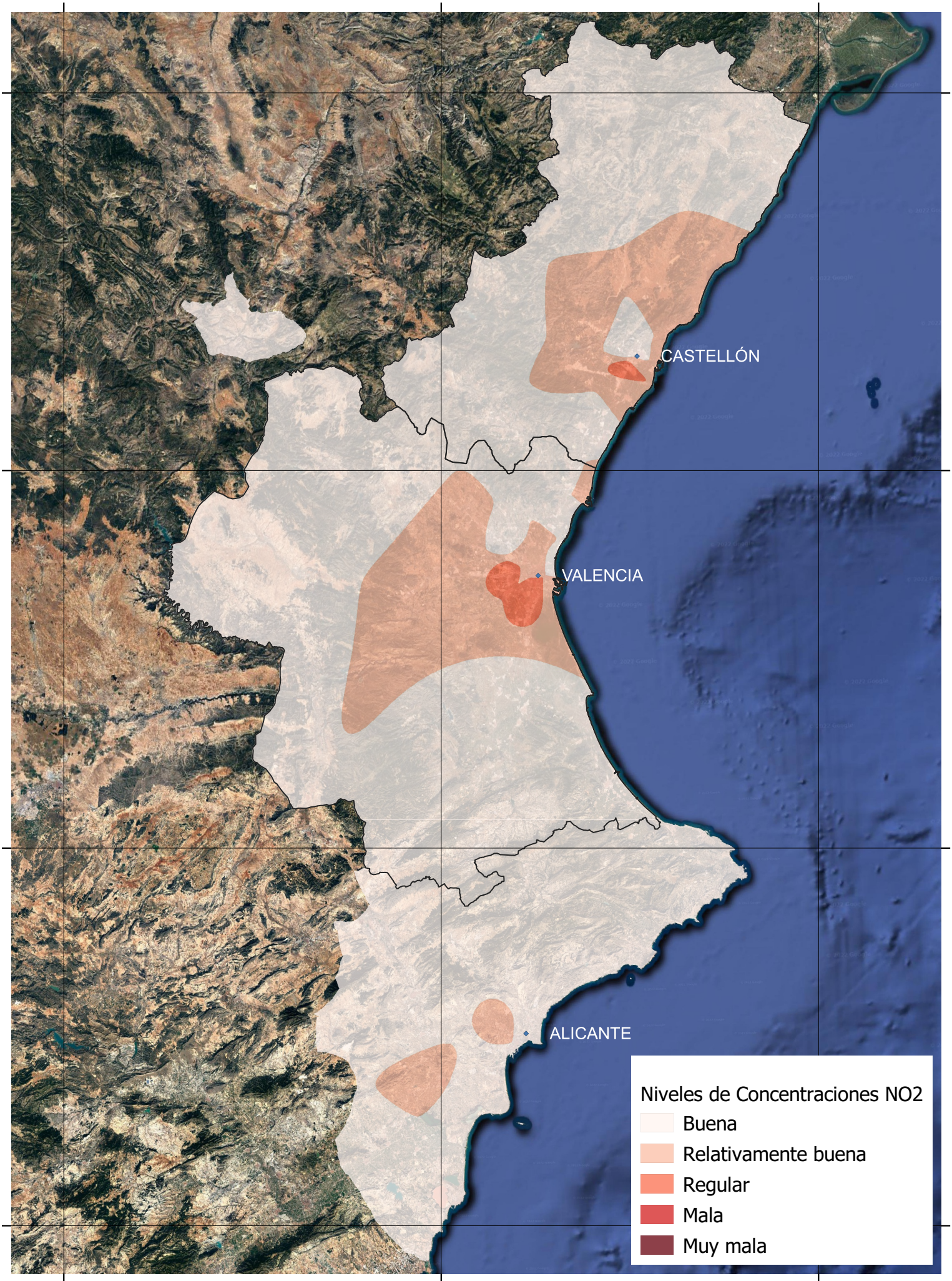


MAPA CONCENTRACIONES NO2 MARZO 2021 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

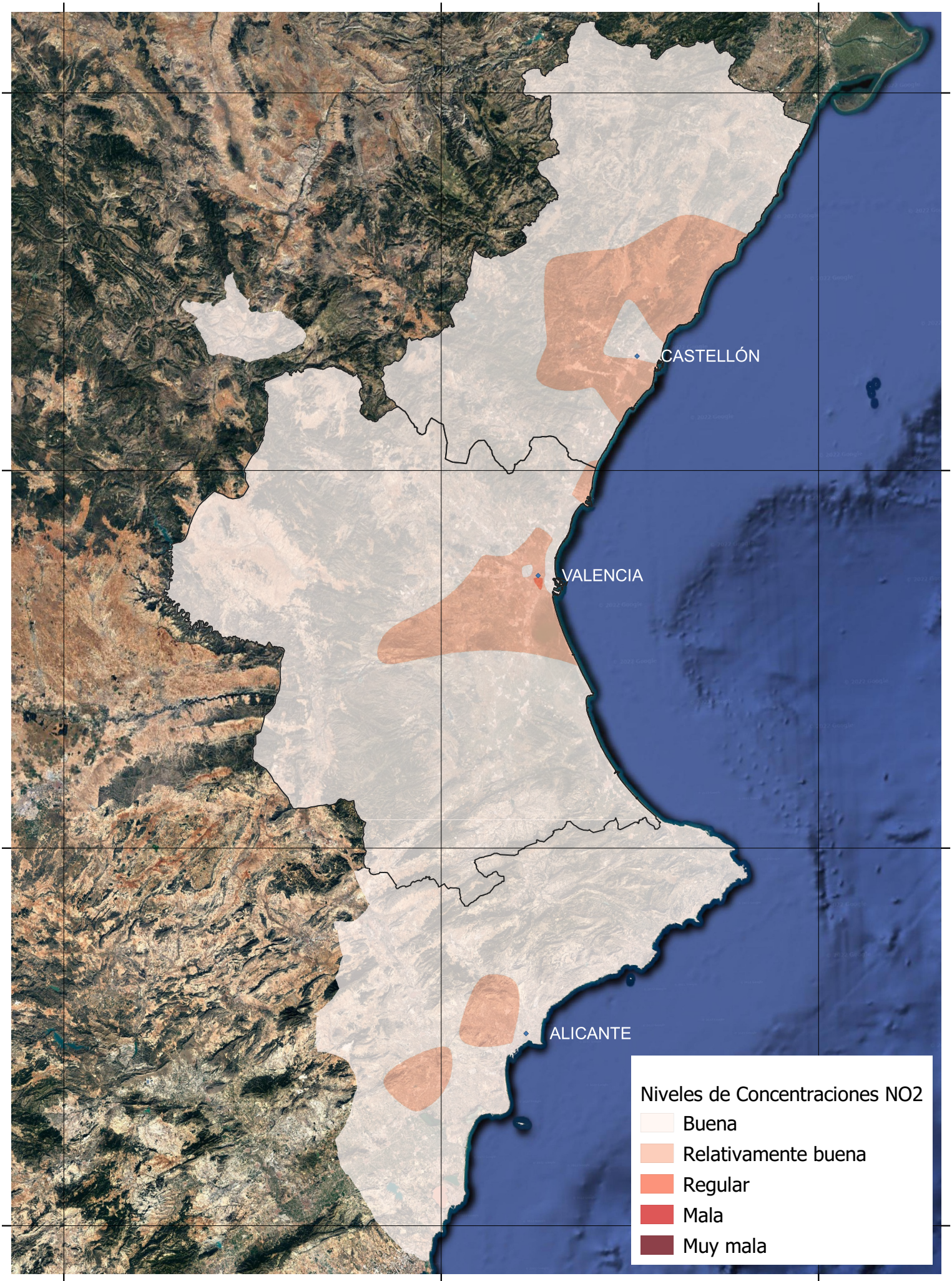


MAPA CONCENTRACIONES NO2 ABRIL 2021 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA



MAPA CONCENTRACIONES NO2 MAYO 2021 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

MAPA CONCENTRACIONES NO2 JUNIO 2021 COMUNIDAD VALENCIANA

0 10 20 30 40 50 km

ESCALA 1:1.250.000 ETRS89 / UTM ZONA 30

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA

10. ANEXOS

10.1 Códigos íntegros

10.1.1 Descarga automática de datos meteorológicos horarios AEMET en la CV

10.1.2 Descarga masiva de datos diarios de AEMET

10.1.3 Descarga masiva de imágenes Sentinel 5P mediante api GEE en Python

10.1.4 Programa python en Jupyter Notebook de cálculo y tratamiento de los datos

10.1.5 Códigos R

Descarga automática de datos meteorológicos horarios AEMET en la CV

```
import http.client
import json
import datetime
import os
import dropbox
import time
import io
import schedule

def ciclo1():
    # CLIENTE DROPBOX

    cliente = dropbox.Dropbox('apidropbox')

    # Estaciones de la Comunidad Valenciana

    estaciones = ['8501', '8500A', '8489X', '8523X', '9563X', '8058X',
'8325X', '8414A', '8416Y', '8309X', '8416',
                  '8293X', '7247X', '8025', '8019', '8050X']
    nomestaciones = ['CASTELLÓ DE LA PLANA', 'CASTELLÓN - ALMASSORA',
'VILLAFRANCA DEL CID', 'VINARÒS', 'CASTELLFORT',
                     'OLIVA', 'POLINYÀ DE XÚQUER', 'VALENCIA
AEROPUERTO', 'VALÈNCIA, VIVEROS', 'UTIEL', 'VALÈNCIA',
                     'XÀTIVA', 'PINOSO', 'ALICANTE', 'ALICANTE-ELCHE
AEROPUERTO', 'JÁVEA']
    #CONTADOR

    contador = 0

    # FECHA

    fecha = datetime.date.today().strftime("%b-%d-%Y")

    # Indicamos la carpeta donde alojar los datos en dropbox

    carpetaraiz = '/' + fecha + '_Datos_Ciclo1/'
    carpetameta = carpetaraiz + fecha + '_Metadatos/'
    carpetajson = carpetaraiz + fecha + '_Datos/'
    carpetainforme = carpetaraiz + fecha + '_Informe/'

    # CONEXIÓN AEMET

    conn = http.client.HTTPSConnection("opendata.aemet.es")

    headers = {
        'cache-control': "no-cache"
    }

    # API KEY

    perapikey = "apikey"

    for i in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = i
```

```
            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_k
ey={perapikey}",
                         headers=headers)

            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldatos = datajson['datos']

            # PATHS

            ficherocompleto = os.path.join(carpetajson, fecha +
"_CLIMA_HORARIO_CV_" + nombreest + ".txt")

            # JSON

            jsontxt = cliente.files_save_url(ficherocompleto,
urldatos)

            # PARÁMETROS

            contador = contador + 1
            time.sleep(10)

        except:
            txt = "No hay datos que satisfagan esos criterios Estado
404"

            with io.BytesIO(txt.encode()) as stream:
                stream.seek(0)

                # Write a text file
                cliente.files_upload(stream.read(), carpetainforme +
f"{fecha}_{nombreest}_informe.txt",

mode=dropbox.files.WriteMode.overwrite)

            contador = contador + 1
            time.sleep(10)
            pass

    contador = 0

    for j in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = j

            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_k
ey={perapikey}",
                         headers=headers)
```

```
            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldmetadatos = datajson["metadatos"]

            # PATHS

            ficherocompletometadatos = os.path.join(carpetameta,
                                                    fecha +
"_CLIMA_HORARIO_CV_" + nombreest + "_metadatos.txt")
            ficherocompletoinforme_meta = os.path.join(carpetainforme,
                                                       fecha +
"_InformeMetadatos_" + nombreest + ".txt")

            # JSON

            metadatos =
cliente.files_save_url(ficherocompletometadatos, urldmetadatos)

            # PARÁMETROS

            contador = contador + 1
            time.sleep(10)


        except:
            contador = contador + 1
            time.sleep(10)
            pass

def ciclo2():
    # CLIENTE DROPBOX

    cliente = dropbox.Dropbox('apidropbox')

    # Estaciones de la Comunidad Valenciana

    estaciones = ['8501', '8500A', '8489X', '8523X', '9563X', '8058X',
'8325X', '8414A', '8416Y', '8309X', '8416',
                  '8293X', '7247X', '8025', '8019', '8050X']
    nomestaciones = ['CASTELLÓ DE LA PLANA', 'CASTELLÓN - ALMASSORA',
'VILLAFRANCA DEL CID', 'VINARÒS', 'CASTELLFORT',
                     'OLIVA', 'POLINYÀ DE XÚQUER', 'VALENCIA
AEROPUERTO', 'VALÈNCIA, VIVEROS', 'UTIEL', 'VALÈNCIA',
                     'XÀTIVA', 'PINOSO', 'ALICANTE', 'ALICANTE-ELCHE
AEROPUERTO', 'JÁVEA']

    #CONTADOR

    contador = 0

    # FECHA

    fecha = datetime.date.today().strftime("%b-%d-%Y")

    # Indicamos la carpeta donde alojar los datos en dropbox

    carpetaraiz = '/' + fecha + '_Datos_Ciclo2/'
    carpetameta = carpetaraiz + fecha + '_Metadatos/'
```

```
    carpetajson = carpetaraiz + fecha + '_Datos/'
    carpetainforme = carpetaraiz + fecha + '_Informe/'

    # CONEXIÓN AEMET

    conn = http.client.HTTPSConnection("opendata.aemet.es")

    headers = {
        'cache-control': "no-cache"
    }

    # API KEY

    perapikey = "apikey"

    for i in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = i

            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_k
ey={perapikey}",
                         headers=headers)

            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldatos = datajson['datos']
            descripcion = datajson['descripcion']

            # PATHS

            ficherocompleto = os.path.join(carpetajson, fecha +
"_CLIMA_HORARIO_CV_" + nombreest + ".txt")

            # JSON

            jsontxt = cliente.files_save_url(ficherocompleto,
urldatos)

            # PARÁMETROS

            contador = contador + 1
            time.sleep(10)

        except:
            txt = "No hay datos que satisfagan esos criterios Estado
404"

            with io.BytesIO(txt.encode()) as stream:
                stream.seek(0)

                # Write a text file
                cliente.files_upload(stream.read(), carpetainforme +
f"{fecha}_{nombreest}_informe.txt",
```

```
mode=dropbox.files.WriteMode.overwrite)

            contador = contador + 1
            time.sleep(10)
            pass

    contador = 0

    for j in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = j

            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_k
ey={perapikey}",
                         headers=headers)

            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldmetadatos = datajson["metadatos"]

            # PATHS

            ficherocompletometadatos = os.path.join(carpetameta,
                                                    fecha +
"_CLIMA_HORARIO_CV_" + nombreest + "_metadatos.txt")
            ficherocompletoinforme_meta = os.path.join(carpetainforme,
                                                       fecha +
"_InformeMetadatos_" + nombreest + ".txt")

            # JSON

            metadatos =
cliente.files_save_url(ficherocompletometadatos, urldmetadatos)

            # PARÁMETROS

            contador = contador + 1
            time.sleep(10)


        except:
            contador = contador + 1
            time.sleep(10)
            pass

schedule.every().day.at("08:00").do(ciclo1)
schedule.every().day.at("20:00").do(ciclo2)

while True:
    schedule.run_pending()
    time.sleep(1)
```

Keep Alive Function

```
from flask import Flask
from threading import Thread

app=Flask('')

@app.route('/')
def home():
  return "Hello, i am alive!"

def run():
  app.run(host='0.0.0.0',port=8080)

def keep_alive():
  t=Thread(target=run)
  t.start()
```

Código principal en Replit

```
from keep_alive import keep_alive

keep_alive()

import http.client
import json
import datetime
import os
import dropbox
import time
import  io
import schedule

def ciclo1():
    # CLIENTE DROPBOX
    apidrop= os.environ['apidropbox']
```

```
    cliente = dropbox.Dropbox(apidrop)


    # Estaciones de la Comunidad Valenciana


    estaciones = ['8501', '8500A', '8489X', '8523X', '9563X', '8058X', '8325X', '8414A', '8416Y',
'8309X', '8416',
                  '8293X', '7247X', '8025', '8019', '8050X']
    nomestaciones = ['CASTELLÓ DE LA PLANA', 'CASTELLÓN - ALMASSORA',
'VILLAFRANCA DEL CID', 'VINARÒS', 'CASTELLFORT',
                     'OLIVA', 'POLINYÀ DE XÚQUER', 'VALENCIA AEROPUERTO',
'VALÈNCIA, VIVEROS', 'UTIEL', 'VALÈNCIA',
                     'XÀTIVA', 'PINOSO', 'ALICANTE', 'ALICANTE-ELCHE AEROPUERTO',
'JÁVEA']


    contador = 0


    # FECHA


    fecha = datetime.date.today().strftime("%b-%d-%Y")


    # Indicamos la carpeta donde alojar los datos en dropbox


    carpetaraiz = '/' + fecha + '_Datos_Ciclo1/'
    carpetameta = carpetaraiz + fecha + '_Metadatos/'
    carpetajson = carpetaraiz + fecha + '_Datos/'
    carpetainforme = carpetaraiz + fecha + '_Informe/'


    # CONEXIÓN AEMET


    conn = http.client.HTTPSConnection("opendata.aemet.es")


    headers = {
       'cache-control': "no-cache"
    }
```

```
    # API KEY

    apiclima = os.environ['apiaemet']
    perapikey = apiclima

    for i in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = i

            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_key={perapikey}",
                         headers=headers)

            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldatos = datajson['datos']
            descripcion = datajson['descripcion']

            # PATHS

            ficherocompleto = os.path.join(carpetajson, fecha + "_CLIMA_HORARIO_CV_" +
nombreest + ".txt")

            # JSON

            jsontxt = cliente.files_save_url(ficherocompleto, urldatos)
```

```
        # PARÁMETROS

        contador = contador + 1
        time.sleep(10)

    except:
        txt = "No hay datos que satisfagan esos criterios Estado 404"

        with io.BytesIO(txt.encode()) as stream:
            stream.seek(0)

            # Write a text file
            cliente.files_upload(stream.read(), carpetainforme +
f"{fecha}_{nombreest}_informe.txt",
                                 mode=dropbox.files.WriteMode.overwrite)

        contador = contador + 1
        time.sleep(10)
        pass

contador = 0

for j in estaciones:
    try:
        nombreest = nomestaciones[contador]
        actual = j

        # PETICIÓN

        conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_key={perapikey}",
                     headers=headers)
```

```
            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldmetadatos = datajson["metadatos"]

            # PATHS

            ficherocompletometadatos = os.path.join(carpetameta,
                                        fecha + "_CLIMA_HORARIO_CV_" + nombreest +
"_metadatos.txt")
            ficherocompletoinforme_meta = os.path.join(carpetainforme,
                                          fecha + "_InformeMetadatos_" + nombreest + ".txt")

            # JSON

            metadatos = cliente.files_save_url(ficherocompletometadatos, urldmetadatos)

            # PARÁMETROS

            contador = contador + 1
            time.sleep(10)



        except:
            contador = contador + 1
            time.sleep(10)
            pass

def ciclo2():
    # CLIENTE DROPBOX
```

```
    apidrop= os.environ['apidropbox']
    cliente = dropbox.Dropbox(apidrop)

    # Estaciones de la Comunidad Valenciana

    estaciones = ['8501', '8500A', '8489X', '8523X', '9563X', '8058X', '8325X', '8414A', '8416Y',
'8309X', '8416',
                  '8293X', '7247X', '8025', '8019', '8050X']
    nomestaciones = ['CASTELLÓ DE LA PLANA', 'CASTELLÓN - ALMASSORA',
'VILLAFRANCA DEL CID', 'VINARÒS', 'CASTELLFORT',
                     'OLIVA', 'POLINYÀ DE XÚQUER', 'VALENCIA AEROPUERTO',
'VALÈNCIA, VIVEROS', 'UTIEL', 'VALÈNCIA',
                     'XÀTIVA', 'PINOSO', 'ALICANTE', 'ALICANTE-ELCHE AEROPUERTO',
'JÁVEA']

    contador = 0

    # FECHA

    fecha = datetime.date.today().strftime("%b-%d-%Y")

    # Indicamos la carpeta donde alojar los datos en dropbox

    carpetaraiz = '/' + fecha + '_Datos_Ciclo2/'
    carpetameta = carpetaraiz + fecha + '_Metadatos/'
    carpetajson = carpetaraiz + fecha + '_Datos/'
    carpetainforme = carpetaraiz + fecha + '_Informe/'

    # CONEXIÓN AEMET

    conn = http.client.HTTPSConnection("opendata.aemet.es")

    headers = {
        'cache-control': "no-cache"
```

```
    }

    # API KEY
    apiclima = os.environ['apiaemet']
    perapikey = apiclima

    for i in estaciones:
        try:
            nombreest = nomestaciones[contador]
            actual = i

            # PETICIÓN

            conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_key={perapikey}",
                         headers=headers)

            # RESPUESTA Y URL

            res = conn.getresponse()
            data = res.read()
            datajson = json.loads(data)
            urldatos = datajson['datos']
            descripcion = datajson['descripcion']

            # PATHS

            ficherocompleto = os.path.join(carpetajson, fecha + "_CLIMA_HORARIO_CV_" +
nombreest + ".txt")

            # JSON

            jsontxt = cliente.files_save_url(ficherocompleto, urldatos)
```

```
        # PARÁMETROS

        contador = contador + 1
        time.sleep(10)

    except:
        txt = "No hay datos que satisfagan esos criterios Estado 404"

        with io.BytesIO(txt.encode()) as stream:
            stream.seek(0)

            # Write a text file
            cliente.files_upload(stream.read(), carpetainforme +
f"{fecha}_{nombreest}_informe.txt",
                                 mode=dropbox.files.WriteMode.overwrite)

        contador = contador + 1
        time.sleep(10)
        pass

contador = 0

for j in estaciones:
    try:
        nombreest = nomestaciones[contador]
        actual = j

        # PETICIÓN

        conn.request(f"GET",
f"/opendata/api/observacion/convencional/datos/estacion/{actual}?api_key={perapikey}",
                     headers=headers)
```

```
        # RESPUESTA Y URL

        res = conn.getresponse()
        data = res.read()
        datajson = json.loads(data)
        urldmetadatos = datajson["metadatos"]

        # PATHS

        ficherocompletometadatos = os.path.join(carpetameta,
                                    fecha + "_CLIMA_HORARIO_CV_" + nombreest +
"_metadatos.txt")
        ficherocompletoinforme_meta = os.path.join(carpetainforme,
                                      fecha + "_InformeMetadatos_" + nombreest + ".txt")

        # JSON

        metadatos = cliente.files_save_url(ficherocompletometadatos, urldmetadatos)

        # PARÁMETROS

        contador = contador + 1
        time.sleep(10)



    except:
        contador = contador + 1
        time.sleep(10)
        pass

schedule.every().day.at("07:00").do(ciclo1)
schedule.every().day.at("19:00").do(ciclo2)
```

```
while True:
    schedule.run_pending()
    time.sleep(1)
```

Descarga masiva de datos diarios de AEMET

```
import http.client
import json
import time
import urllib.request
import datetime
import os
import pandas as pd
import codecs
import csv
import glob

#Estaciones de la Comunidad Valenciana

estaciones=['8501','8500A','8489X','8523X','9563X','8058X','8325X','8414A','8416Y','8309X'
nomestaciones=['CASTELLÓ DE LA PLANA','CASTELLÓN - ALMASSORA','VILLAFRANCA DEL CID','VINAR

contador=0
porcentaje=100/len(estaciones)

#Indicamos las fechas

fechaini=str(input("Indique la fecha inicial con este formato aaaa-mm-dd: "))
fechafin=str(input("Indique la fecha final con este formato aaaa-mm-dd: "))

#Indicamos la carpeta donde alojar los datos

carpeta=r'C:\\Users\\Usuario\\Desktop\\CSVS\\aemet_diarioCV\\datos\\'
carpetameta=r'C:\\Users\\Usuario\\Desktop\\CSVS\\aemet_diarioCV\\metadatos\\'
carpetainforme=r'C:\\Users\\Usuario\\Desktop\\CSVS\\aemet_diarioCV\\informes\\'

#CONEXIÓN AEMET

conn = http.client.HTTPSConnection("opendata.aemet.es")

headers = {
    'cache-control': "no-cache"
    }

#API KEY

perapikey="apikey"

for i in estaciones:
    nombreest = nomestaciones[contador]
    actual = i

    # PETICIÓN

    conn.request(f"GET",
```

```
            f"/opendata/api/valores/climatologicos/diarios/datos/fechaini/{fechaini}T
            headers=headers)
res = conn.getresponse()
data = res.read()
datajson = json.loads(data)
try:
    # RESPUESTA Y URL

    urldatos = datajson["datos"]
    urlmetadatos = datajson["metadatos"]

    #PATHS

    ficherocompleto = os.path.join(carpeta,
                                   fechaini + "a" + fechafin + "_CLIMATOLOGÍA_DIARIA_C
    ficherocompletometa = os.path.join(carpetameta,
                                       fechaini + "a" + fechafin + "_CLIMATOLOGÍA_DIAR
    ficherocompletoutf8 = os.path.join(carpeta,
                                       fechaini + "a" + fechafin + "_CLIMATOLOGÍA_DIAR
    ficherocompletocsv = os.path.join(carpeta,
                                      fechaini + "a" + fechafin + "_CLIMATOLOGÍA_DIARI
    ficherocompletocsvutf8 = os.path.join(carpeta,
                                          fechaini + "a" + fechafin + "_CLIMATOLOGÍA_D

    # JSON Y METADATOS

    jsontxt = urllib.request.urlretrieve(urldatos, ficherocompleto)
    metadatostxt = urllib.request.urlretrieve(urlmetadatos, ficherocompletometa)

    with codecs.open(ficherocompleto, "r", "ANSI") as sourceFile:
        with codecs.open(ficherocompletoutf8, "w", "utf-8") as targetFile:
            while True:
                contents = sourceFile.read()
                if not contents:
                    break
                targetFile.write(contents)

    # CSV

    df = pd.read_json(ficherocompletoutf8)
    df.to_csv(ficherocompletocsv, index=None, encoding='utf-8-sig')

    os.remove(ficherocompleto)
    os.remove(ficherocompletoutf8)

    print(f"Se ha descargado la estación {nombreest}, descarga al {porcentaje}%")

    contador = contador + 1
    porcentaje = porcentaje + (100 / len(estaciones))

    time.sleep(10)
except:
    # RESPUESTA Y URL

    descripcion=datajson["descripcion"]
```

```
        # PATHS

        ficherocompletoinformes = os.path.join(carpetainforme,fechaini + "a" + fechafin +

        #Informe

        with open(ficherocompletoinformes, 'w') as f:
            f.write(f"La estación {nombreest} no puede descargar los datos debido a: {desc

        print(f"La estación {nombreest} no puede descargar los datos debido a: {descripcio

        contador=contador+1
        porcentaje=porcentaje + (100 / len(estaciones))

        time.sleep(10)

#JUNTAR CSVS

os.chdir(carpeta)
extension="csv"
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]
combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])
combined_csv.to_csv( f"{fechaini}a{fechafin}_CLIMATOLOGÍA_DIARÍA_CV.csv", index=False, enc
```

Descarga masiva de imágenes Sentinel 5P mediante api GEE en Python

```
import ee

# Trigger the authentication flow.
ee.Authenticate()

# Initialize the library.
ee.Initialize()


CV =ee.Geometry.Polygon([[[-1.5529172193195917, 40.84159458236475],
          [-1.5529172193195917, 37.74763117383071],
          [0.6882937181804083, 37.74763117383071],
          [0.6882937181804083, 40.84159458236475]]])

START = '2021-01-01'
END = '2022-01-01'

producto = ee.ImageCollection("COPERNICUS/S5P/OFFL/L3_O3").filter(ee.Filter.date(START, EN

collectionList = producto.toList(producto.size())
collectionSize = collectionList.size().getInfo()
for i in range(collectionSize):
    fecha=str(ee.Image(collectionList.get(i)).get('system:index').getInfo())[0:15]
    reducer = ee.Reducer.max()
    imageni= ee.Image(collectionList.get(i)).select('O3_column_number_density').clip(CV)
    cuentaPixelesFunction= imageni.reduceRegion(reducer,CV,1000)
    cuentaPixeles = str(cuentaPixelesFunction.get('O3_column_number_density').getInfo())
    if cuentaPixeles!= 'None':
      print('O3_'+fecha+'_Píxeles:'+str(cuentaPixeles))
      ee.batch.Export.image.toDrive(
        image = ee.Image(collectionList.get(i)).clip(CV),
        folder='O3_2021',
        description = 'O3_'+fecha,
        fileNamePrefix = 'O3_'+fecha,
        scale = 1000).start()
    else:
      pass
```

Programa python en Jupyter Notebook de cálculo y tratamiento de los datos

```
import pandas as pd
import numpy as np

df = pd.read_csv('RVVCCA_CASTELLON_CSV.csv',encoding= 'unicode_escape',low_memory=False)

df['FECHA (DD/MM/YYYY)']= df['FECHA (DD/MM/YYYY)'].replace('/','',regex=True)

df['FECHA (DD/MM/YYYY)'] = pd.to_datetime(df['FECHA (DD/MM/YYYY)'],format='%d%m%Y')

listaEstaciones=['12009007-Almassora - C.P.Ochando','12040302-Castello AP Gregal','1204030

lat=['39.94420428','39.96972214','39.96667768','39.96305554','39.97499992','39.95025201','
lon=['-0.05739487','0.01361108','0.00890491','0.00694444','0.01638888','0.00501868','0.071

cont=0

for estacion in listaEstaciones:
    dfF2020=df.loc[(df['FECHA (DD/MM/YYYY)']>='2020-02-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfM2020=df.loc[(df['FECHA (DD/MM/YYYY)']>='2020-03-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfA2020=df.loc[(df['FECHA (DD/MM/YYYY)']>='2020-04-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfMa2020=df.loc[(df['FECHA (DD/MM/YYYY)']>='2020-05-01') & (df['FECHA (DD/MM/YYYY)']<=
    dfJ2020=df.loc[(df['FECHA (DD/MM/YYYY)']>='2020-06-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfF2021=df.loc[(df['FECHA (DD/MM/YYYY)']>='2021-02-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfM2021=df.loc[(df['FECHA (DD/MM/YYYY)']>='2021-03-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfA2021=df.loc[(df['FECHA (DD/MM/YYYY)']>='2021-04-01') & (df['FECHA (DD/MM/YYYY)']<='
    dfMa2021=df.loc[(df['FECHA (DD/MM/YYYY)']>='2021-05-01') & (df['FECHA (DD/MM/YYYY)']<=
    dfJ2021=df.loc[(df['FECHA (DD/MM/YYYY)']>='2021-06-01') & (df['FECHA (DD/MM/YYYY)']<='
    with open(f'{estacion}.csv', 'w') as f:
        f.write('Estacion,LAT,LON,FO320,FNO220,MO320,MNO220,AO320,ANO220,MaO320,MaNO220,JO
        f.write(f'{estacion},{lat[cont]},{lon[cont]},{round(dfF2020["O3 (µg/m³)"].astype(f
    cont=cont+1
```

CÓDIGO R HISTOGRAMA FRECUENCIA

```
##TFG=group
##showplots
##Layer=vector
##Parameter=Field Layer
##Number_bins= number
##Label_x= string
##Label_y= string
##Title_graph= string
##Subtitle_graph= string
library(ggplot2)

ggplot(NULL, aes(Layer[[Parameter]]))+
geom_histogram(bins=Number_bins, col=1, fill=6,
alpha=.2,binwidth=1,center = 1) +
scale_y_continuous(breaks = seq(0, 12, by = 2))+
scale_x_continuous(breaks = seq(0, 40, by = 5))+
expand_limits(y = c(0, 12))+
expand_limits(x = c(1, 40))+
labs(x= Label_x,y=Label_y, title = Title_graph,
subtitle=Subtitle_graph) +
theme(plot.title = element_text( face = "bold",size = 30,hjust =0.5,
color = "Black")) +
theme(axis.text = element_text(colour = "black", size =10, face =
"bold")) +
theme(plot.subtitle=element_text(size=12, hjust=0.5, face="italic",
color="black"))
```

CÓDIGO R HISTOGRAMA FRECUENCIA RELATIVA

```
##TFG=group
##showplots
##Layer=vector
##Parameter=Field Layer
##Label_x= string
##Label_y= string
##Title_graph= string
##Subtitle_graph= string
library(ggplot2)

ggplot(NULL, aes(Layer[[Parameter]])) +
stat_ecdf(aes(colour=Parameter), col="Blue", size=1)+
labs(x= Label_x,y=Label_y, title = Title_graph,
subtitle=Subtitle_graph) +
scale_x_continuous(breaks = seq(0, 40, by = 5))+
expand_limits(x = c(1, 40))+
theme(plot.title = element_text( face = "bold",size = 20,hjust =0.5,
color = "Black")) +
theme(axis.text = element_text(colour = "black", size =10, face =
"bold")) +
theme(plot.subtitle=element_text(size=12, hjust=0.5, face="italic",
color="Black"))
```