CONTAMINACIÓN EN VALENCIA, INFLUENCIA DE LAS FALLAS Y POSIBLES SOLUCIONES

Pablo López Domínguez

Daniel Macho Collado

Juan Gómez Díaz

Víctor Cardama Noguera

Eran Doni

Visualización de datos (2024), 2º Grado Ciencia de Datos

Índice

CONTENIDOS

1. Introducción

En los últimos años la contaminación se ha convertido en una de las grandes preocupaciones de la sociedad española, siendo un problema que no podemos ignorar.

Cada año, 4.2 millones de personas mueren como resultado de la exposición continua a la contaminación del aire, según la Organización Mundial de la Salud (OMS). Las enfermedades pulmonares, cardiovasculares y relacionadas con accidentes cerebrovasculares, causadas o agravadas por la contaminación, son las principales causas de estas cifras. Y el 91 % de las personas vive en lugares con una concentración de contaminantes en el aire superior a la recomendada por la OMS, particularmente en regiones con mayor presión urbana e industrial.

Y Valencia no se libra de estas características, por lo que hemos visto interesante representar la contaminación en Valencia para arrojar información sobre nuestra situación y cómo podríamos intentar solucionar los problemas de la contaminación.

Este proyecto lo hacemos con el objetivo de dar al ciudadano valenciano una página web accesible y sencilla para visualizar la evolución de la contaminación del aire, junto con las zonas verdes y las estaciones de bicis de la ciudad. Además de poder con este tipo de visualizaciones obtener una respuesta a la mejora de la contaminación en Valencia.

2. Metodología

2.1. Análisis exploratorio de los datos

En este trabajo, se han utilizado los datasets obligatorios:

- Datos diarios calidad aire 2004-2022

  https://valencia.opendatasoft.com/explore/dataset/rvvcca/information/

  Datos no georreferenciados (ej. Excel, csv). Para cada registro, se sabe la estación de medición desde la que se han tomado los datos. Hay un total de 43.388 registros.

- Estaciones de medida de la contaminación atmosférica:
  https://valencia.opendatasoft.com/explore/dataset/estacions-contaminacio-atmosferiques-estacionescontaminacion-atmosfericas/table/
  Datos georreferenciados (ej. Shapefile, GeoJSON). La capa vectorial es de tipo Point.

- Distritos:
  https://valencia.opendatasoft.com/explore/dataset/districtes-distritos/information/?location=10,39.42291,-0.35395&basemap=e4bf90
  Datos georreferenciados (ej. Shapefile, GeoJSON). La capa vectorial es de tipo Polygon.

También se ha hecho uso de los siguientes datasets:

- ValenBisi Disponibilidad:
  https://valencia.opendatasoft.com/explore/dataset/valenbisi-disponibilitat-valenbisi-dsiponibilidad/table/
  Datos georreferenciados (ej. Shapefile, GeoJSON). La capa vectorial es de tipo Point.

- Zonas verdes (planificación):

https://valencia.opendatasoft.com/explore/dataset/zonas-verdes/table/?disjunctive.nivel3
Datos georreferenciados (ej. Shapefile, GeoJSON). La capa vectorial es de tipo Polygon

2.2. Preprocesado de Datos

Primero de todo se miran los archivos de datos disponibles. En este apartado vamos a trabajar con el csv que contiene “Datos diarios calidad aire 2004-2022”. Esto es un tipo de dato no georeferenciado. También se va a hacer uso de de los datos georeferenciados de “Estaciones de medida de la contaminación atmosférica”.

En el primer vistazo de los datos podemos observar que el csv “rvvcca”, tiene datos de 13 estaciones de medición diferentes:

- Pista de SIlla
- Viveros
- Avda. Francia
- Bulevard Sud
- Valencia Centro
- Moli del Sol
- Politecnico
- Conselleria Mateo
- Nazaret Mateo
- Puerto Moll Trans. Ponent
- Puerto llit antic Turia
- Valencia Olivereta
- Puerto Valencia

Sin embargo, en el dataset georeferenciado de las estaciones de contaminación nos encontramos con estas estaciones:

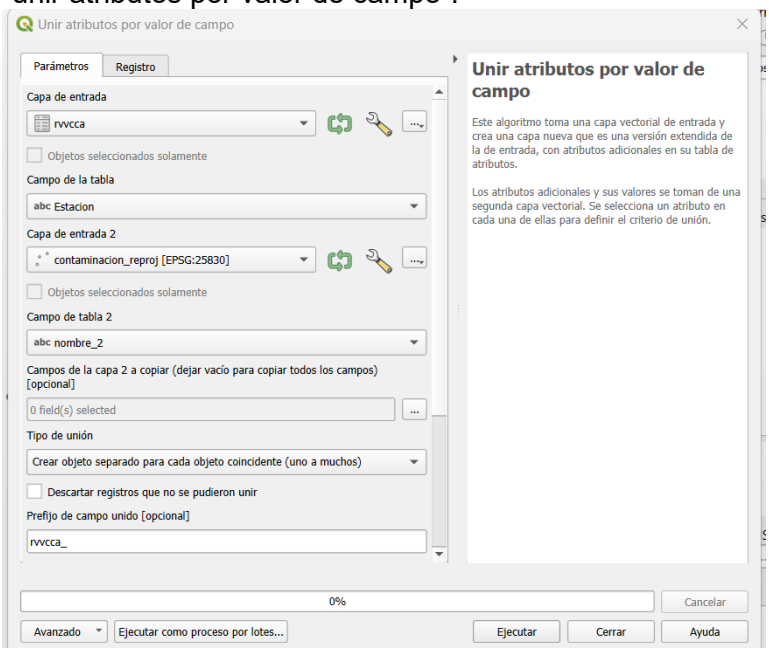
	objectid	nombre	direccion	tipozona	tipoemisio	parametros	mediciones	so2	no2	o3	co	pm10	pm25	fecha_carg
1	432	Patraix	Archiduque Car...	Urbana	Tráfico	Óxidos de nitró...	NULL	NULL	59,0000000000	NULL	NULL	29,0000000000	12,0000000000	2024-04-18 12:20:19+02:00
2	431	Olivereta	Av. del Cid amb...	Urbana	Tráfico	Óxidos de nitró...	NULL	NULL	52,0000000000	NULL	NULL	30,0000000000	11,0000000000	2024-04-18 12:20:18+02:00
3	23	Francia	Avda. de Franci...	Urbana	Tráfico	Dióxido de azuf...	http://mapas.valen...	3,0000000000	12,0000000000	47,0000000000	0,1	16,0000000000	5,0000000000	2024-04-18 12:20:11+02:00
4	25	Moli del Sol	Avda. Pío Baroj...	Suburbana	Tráfico	Dióxido de azuf...	http://mapas.valen...	2,0000000000	33,0000000000	22,0000000000	0,3	9,0000000000	5,0000000000	2024-04-18 12:20:13+02:00
5	829	Cabanyal	Carrer del Prog...	Urbana	Fondo	NULL	NULL	NULL	36,0000000000	NULL	NULL	NULL	NULL	2024-04-18 12:20:18+02:00
6	24	Boulevar Sur	Avda. Tres Cruc...	Urbana	Tráfico	Dióxido de azuf...	http://mapas.valen...	3,0000000000	33,0000000000	11,0000000000	NULL	NULL	NULL	2024-04-18 12:20:12+02:00
7	26	Pista de Silla	C/ Filipinas, s/n	Urbana	Tráfico	Dióxido de azuf...	http://mapas.valen...	6,0000000000	41,0000000000	29,0000000000	0,7	13,0000000000	3,0000000000	2024-04-18 12:20:15+02:00
8	27	Universidad Pol...	Campus de la U...	Suburbana	Fondo	Dióxido de azuf...	http://mapas.valen...	2,0000000000	15,0000000000	20,0000000000	NULL	20,0000000000	10,0000000000	2024-04-18 12:20:15+02:00
9	22	Centro	Plaza Ayuntami...	Urbana	Tráfico	Óxidos de nitró...	https://mapas.vale...	NULL	59,0000000000	NULL	NULL	20,0000000000	8,0000000000	2024-04-18 12:20:17+02:00
10	430	Dr. Lluch	Calle Dr. Lluch, ...	Urbana	Tráfico	Óxidos de nitró...	NULL	NULL	47,0000000000	NULL	NULL	18,0000000000	5,0000000000	2024-04-18 12:20:17+02:00
11	28	Viveros	Jardines de Viv...	Urbana	Fondo	Dióxido de azuf...	http://mapas.valen...	1,0000000000	39,0000000000	29,0000000000	NULL	NULL	NULL	2024-04-18 12:20:16+02:00

Por lo que decidimos comparar los diferentes datasets para unir la locación de las estaciones ya que es un dato georreferenciado, con los datos de las mediciones. Encontramos 8 coincidencias entre el csv y el shp. Una vez elegidas las estaciones a unir, creamos un campo nuevo en el shp, llamado “nombre_2”(todas las capas utilizadas en este proyecto han sido reproyectadas al SCR: EPSG 25830). Una vez se ha creado este campo, se introducen los nombres de las estaciones que aparecen en el csv, con el nombre del csv, ya que los nombres cambian. El campo se crea con la intención de que unir los dos datasets sea una tarea más sencilla, ya que la “clave ” por la que se va a unir va a ser nombre_2, por lo que tiene que coincidir. Así quedaría:

nombre	nombre_2
Patraix	*NULL*
Olivereta	Valencia Olivereta
Francia	Avda. Francia
Molí del Sol	Moli del Sol
Cabanyal	*NULL*
Boulevar Sur	Bulevard Sud
Pista de Silla	Pista Silla
Universidad Politécnica	Politecnico
Centro	Valencia Centro
Dr. Lluch	*NULL*
Viveros	Viveros

Se puede ver que algunos nombres cambian ligeramente (acentos, idioma…), recordamos que “nombre_2” corresponde a los nombres del csv. Con estos datos, pasamos a descartar Patraix, Cabanyal y Dr. LLuch.

Una vez hecho, vamos a crear un shp nuevo llamado “merged_contaminacion” que va a unir las capa de contaminación con el csv. Esto se va a hacer con la herramienta “unir atributos por valor de campo”.



Una vez hecho esto, se nos creará una capa con los atributos de las dos (habiendo descartado los registros que no coinciden). Los atributos procedentes del csv será de esta forma:

rvvcca_Id	rvvcca_Fec	rvvcca_Dia	rvvcca_D_1	rvvcca_Est	rvvcca_PM1

Aquí hemos convertido creado otras columnas con el tipo de dato correspondiente, ya que se copia todo como texto, por lo que hemos creado una columna f_buena, en formato fecha

utilizando la calculadora de campos. Lo mismo para los números, poniéndolos en formato numérico.

f_buena	i_PM1	i_PM2_5	i_PM10	i_NO	i_NO2	i_NOX	i_O3	i_SO2	i_CO

Una vez hecho eso, hemos creado otra columna llamada ICA, para esto hemos usado una fórmula que contempla los límites diarios de algunos marcadores de la contaminación

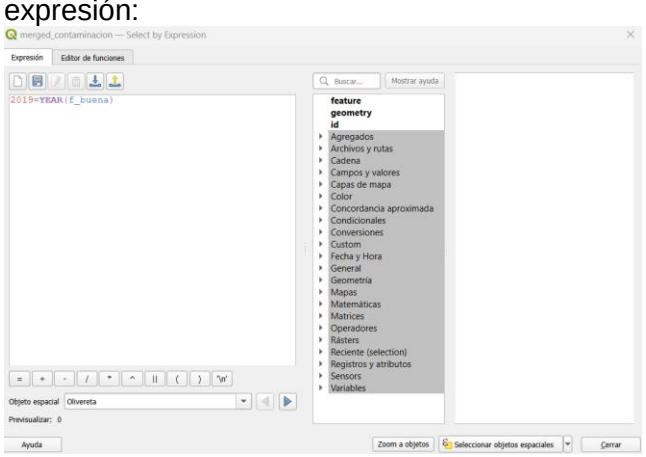
ICA=0.4 (Promedio diario NO2/50) + 0.3(Promedio diario PM 2.525)
+0.3(Promedio diario de 03/120)

Donde los denominadores son los límites diarios de ese marcador para la comunidad valenciana, por lo que si el valor actual pasa del límite, la división será mayor que 1.
Y además se le da un peso diferente a cada marcador.

Una vez creada esa columna, ya se puede tener un indicador de la calidad del aire, cuanto más alto, peor calidad del aire.Al realizar esta operación, cuando falta alguno de los marcadores, el ICA quedará null, para que no de un marcador incorrecto. Esto se ha hecho con la calculadora de campos.

También dividimos el dataset por años, para ver una evolución de la contaminación año a año. (Exceptuando 2018, que a mitad de año una estación más empieza a tomar mediciones).

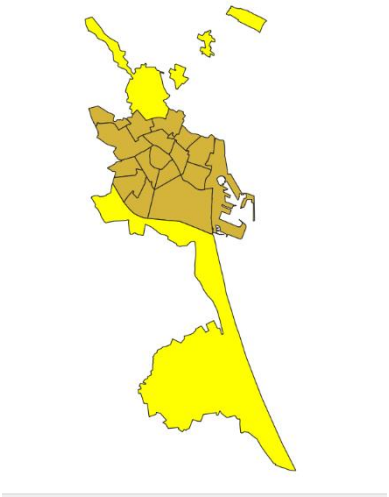
Nuestro estudio se va a enfocar del año 2010 hasta 2022, ya que desde 2004 a 2009 solo hay dos o tres estaciones (dependiendo del año). Por lo que en qgis mediante el selector por expresión, hacemos varios subsets por año. Al ser formato DATE, valdrá con utilizar esta expresión:



Una vez seleccionados se exportan a un nuevo shp mediante exportar objetos seleccionados.

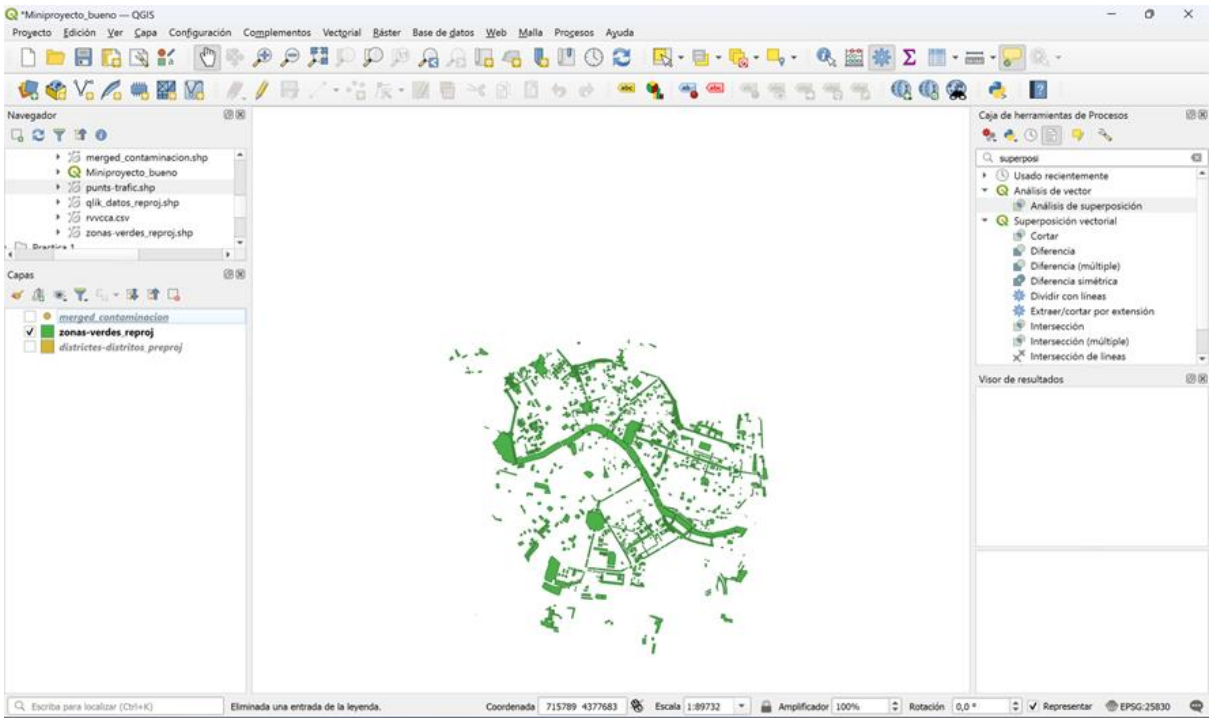
Ahora, pasamos a hacer las interpolaciones. Por la distribución de los datos creemos que la mejor opción para hacer una interpolación del ICA sobre Valencia es la Interpolación IDW. Por conveniencia, pasamos a descartar los distritos de:

- POBLATS DEL NORD
- POBLATS DEL SUD

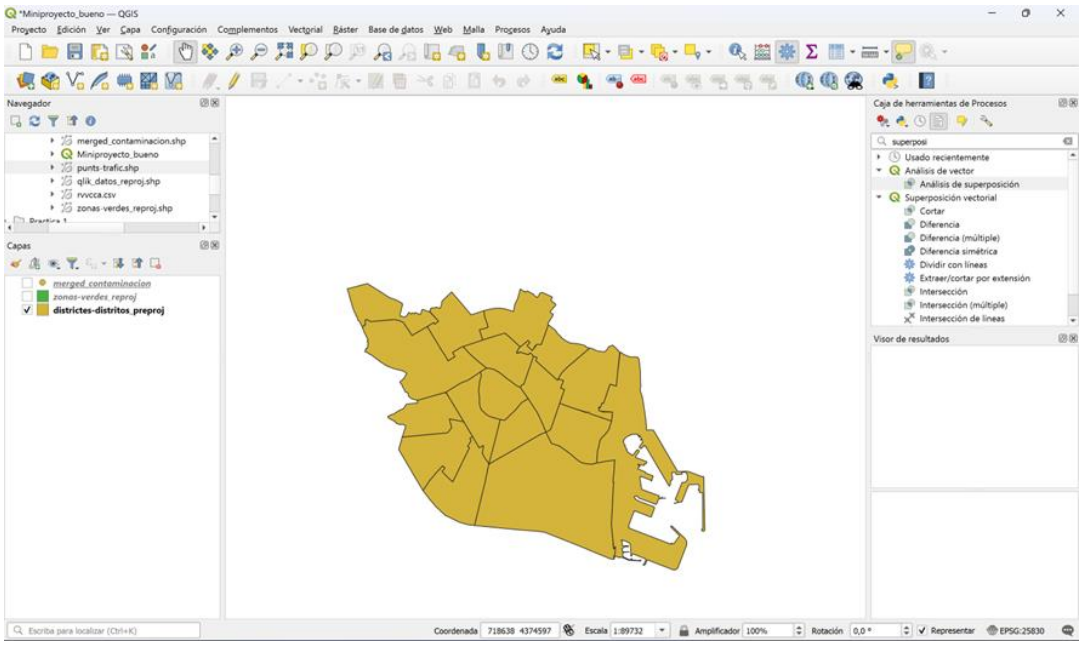


Zonas verdes

Descargamos el archivo con la información sobre las zonas verdes de Valencia disponible en el siguiente. En él, podremos encontrar todas las zonas verdes de Valencia. Una vez descargado lo insertamos en el proyecto y deberemos reproyectar la capa para que este en un SCR correcto, quedando de la siguiente manera:



Una vez hemos reproyectado la capa de zonas verdes tendremos que cargar también la capa de distritos de Valencia, la cual también pondremos en un SRC correcto y reduciremos a los distritos de interés.



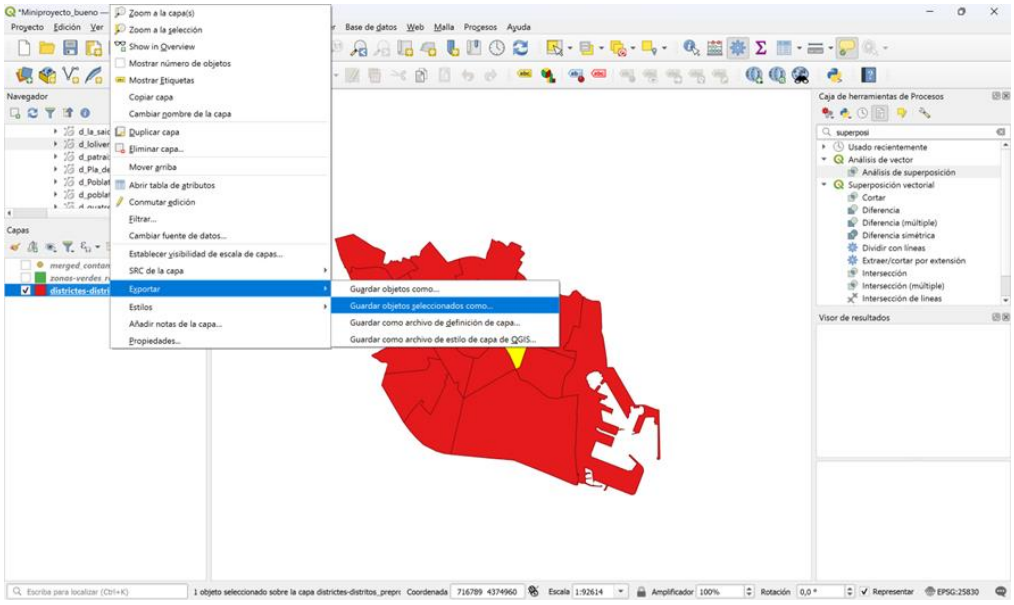
Teniendo ya nuestras capas de interés de forma correcta lo primero que haremos será hacer tantas capas nuevas como distritos tenemos para luego poder seleccionar las zonas verdes de cada distrito. Y quedará de la siguiente manera:

districtes-distritos_preproj— Objetos Totales: 17, Filtrados: 17, Seleccionados: 1

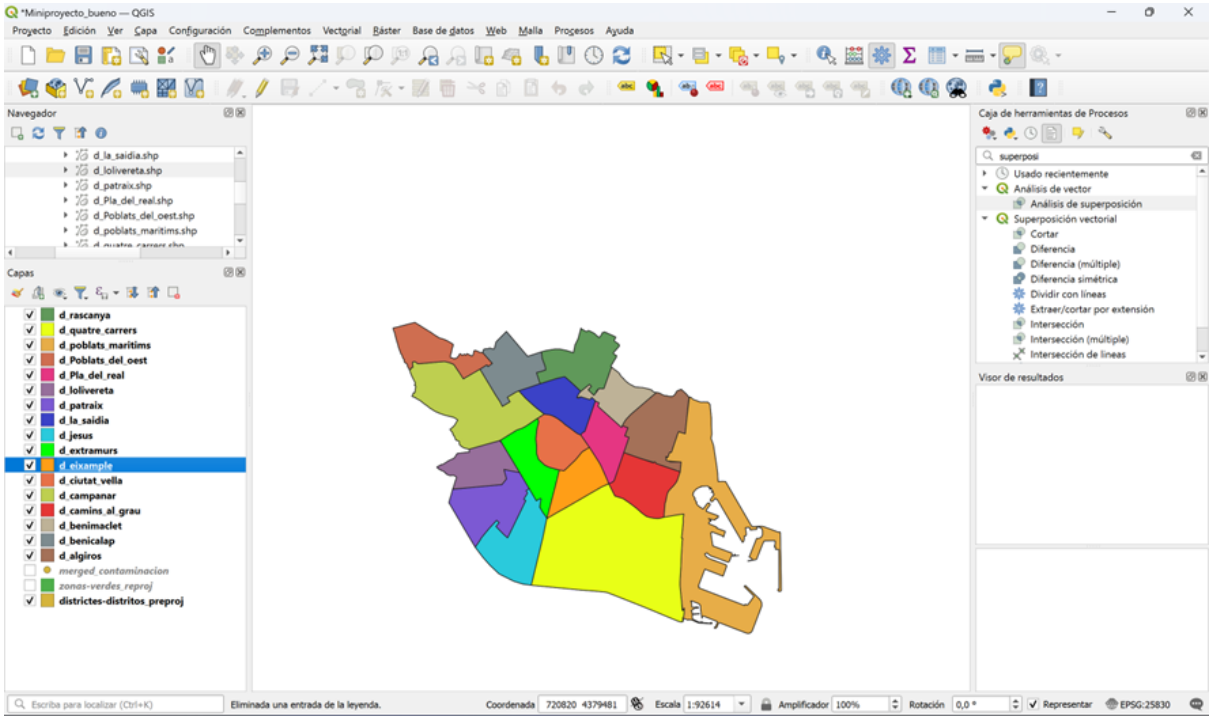
	objectid	nombre	coddistrit	gis_gis_di	latitud	longitud
1	8	EL PLA DEL REAL	6	1692712,87500...	39.4745936913...	-0.3603780274...
2	10	EXTRAMURS	3	1971616,25000...	39.4690241876...	-0.3857800058...
3	12	L'EIXAMPLE	2	1733140,50000...	39.4641145233...	-0.3704393703...
4	71	ALGIROS	13	NULL	39.4776270293...	-0.3427393297...
5	135	RASCANYA	15	NULL	39.4957386280...	-0.3676525156...
6	87	POBLATS DE L'...	18	NULL	39.4999194303...	-0.4177835654...
7	151	L'OLIVERETA	7	NULL	39.4688834120...	-0.4033652971...
8	119	BENICALAP	16	NULL	39.4944929651...	-0.3919111016...
9	9	CIUTAT VELLA	1	1689851,62500...	39.4744154278...	-0.3767596965...
10	14	JESUS	9	2984760,00000...	39.4482014775...	-0.3917921797...
11	1	QUATRE CARRE...	10	NULL	39.4450928959...	-0.3586816784...
12	23	PATRAIX	8	NULL	39.4583977479...	-0.4004125691...
13	6	LA SAIDIA	5	1943931,50000...	39.4850946796...	-0.3750723425...
14	13	CAMINS AL GR...	12	2367473,25000...	39.4627947452...	-0.3478024542...
15	55	BENIMACLET	14	NULL	39.4860570954...	-0.3557092594...
16	103	CAMPANAR	4	NULL	39.4854436586...	-0.4059322619...
17	183	POBLATS MARI...	11	NULL	39.4519525304...	-0.3263356826...

39.469024187643456

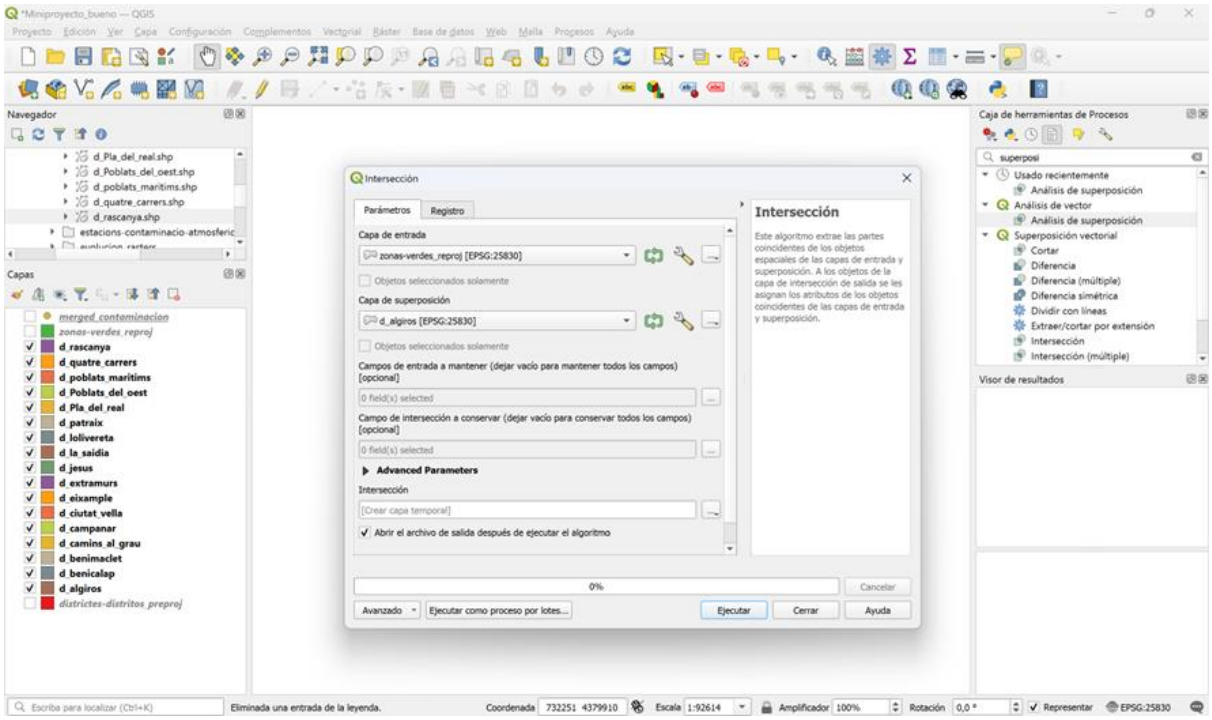
Así uno a uno vamos seleccionando todos los distritos y exportando cada distrito para crear una capa por distrito.



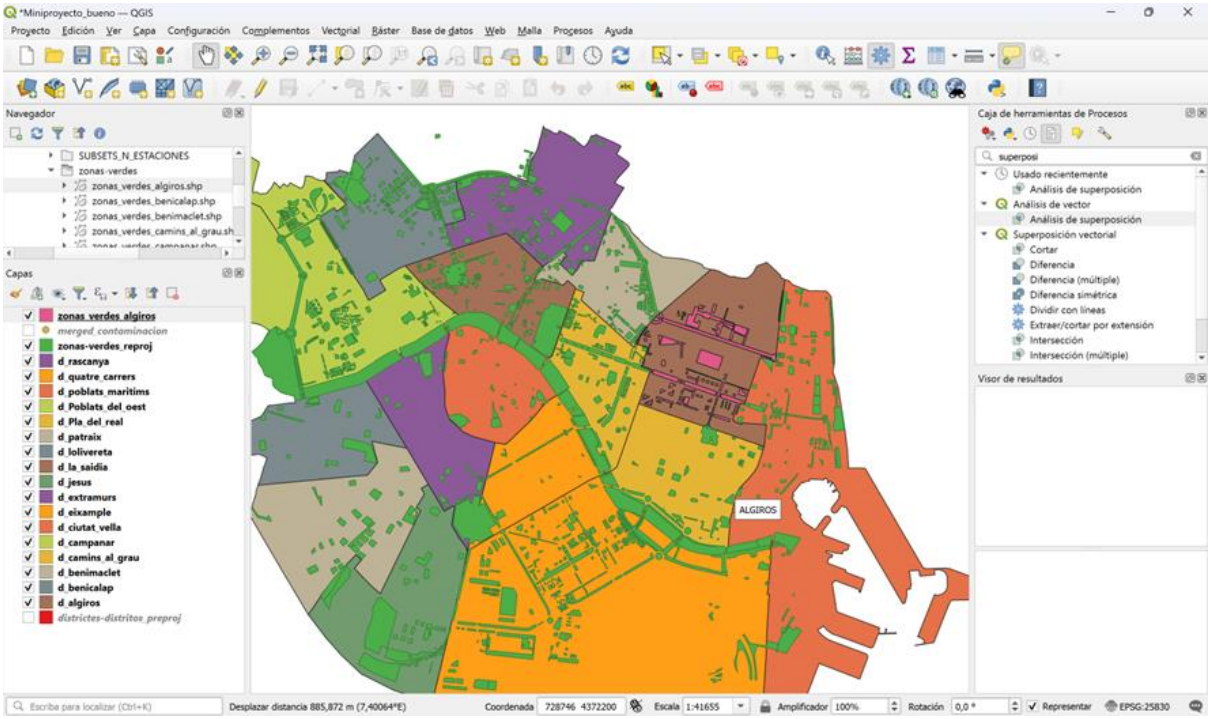
Y una vez todos los distritos exportados uno a uno quedara de la siguiente forma:



Una vez tenemos todos los distritos separados por capas, lo que haremos será trabajar en la capa de zonas verdes realizando una ‘Intersección’ entre la capa reproyectada de zonas verdes y la capa de cada distrito, para crear capas nuevas con las zonas verdes pertenecientes a cada distrito.



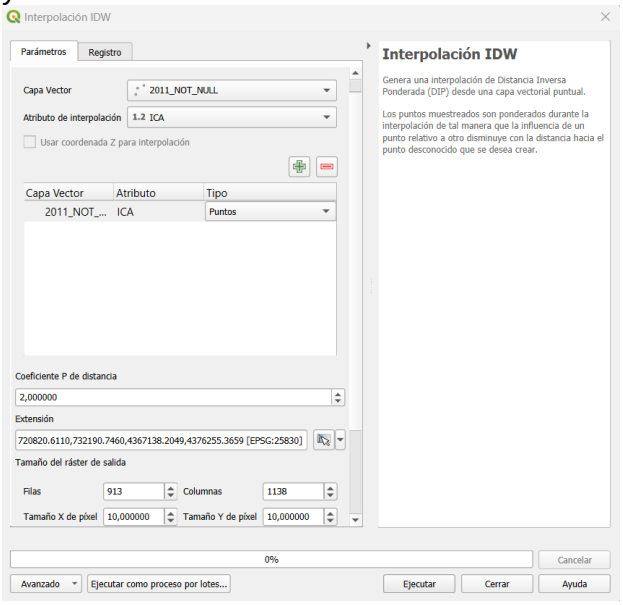
Aquí se ve como se ha hecho la intersección con la capa 'd_algiros:



2.3. Preprocesado de la información geográfica

Para la representación de la evolución de la contaminación año a año:

Ahora pasamos a hacer la interpolación IDW para cada año, se hará con estos parámetros, y considerando una resolución de 10x10.



Una vez se tengan todas las capas rasters, se importarán a r para añadirlas a la aplicación shiny.

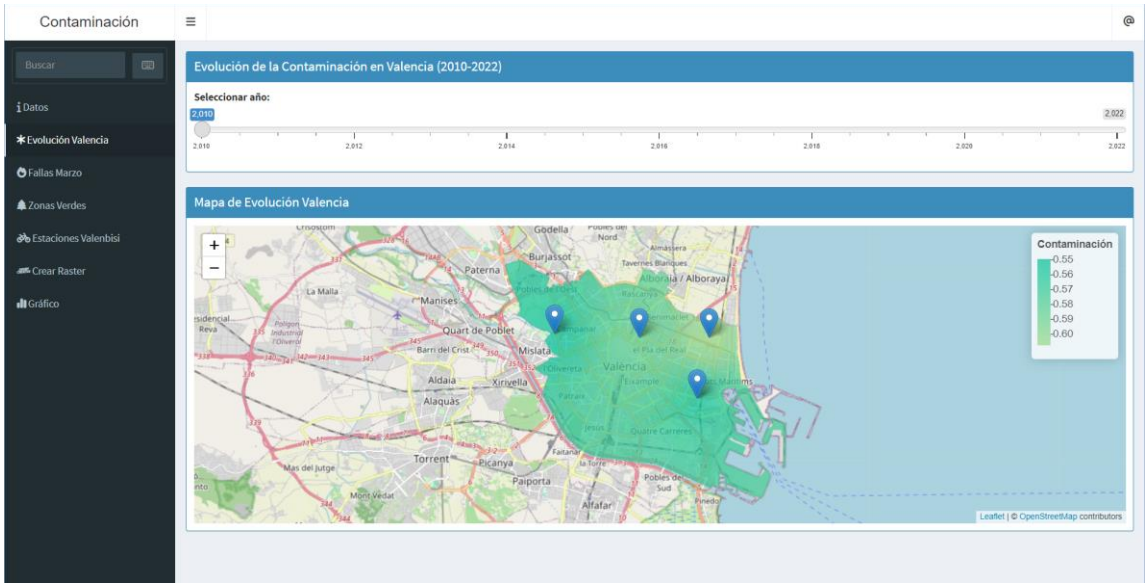
2.4. Elección de gráficas para los distintos tipos de datos

Para la evolución año a año, se ha elegido la opción de visualizar una capa raster, con una misma escala para todos los rasters para así poder comprar un año con otro, independientemente de su valor máximo y mínimo (Por ejemplo: si el valor máximo de 2016 fuera el mínimo de 2017, en un mapa se vería azul y en otro rojo, por lo que no sería comparable a simple vista)

Lo mismo hemos hecho para todos tipos de raster que hemos visualizado en los distintos items (Fallas Marzo, Estaciones Valenbisi, etc.).

2.5. Diseño del mapa e interactividad

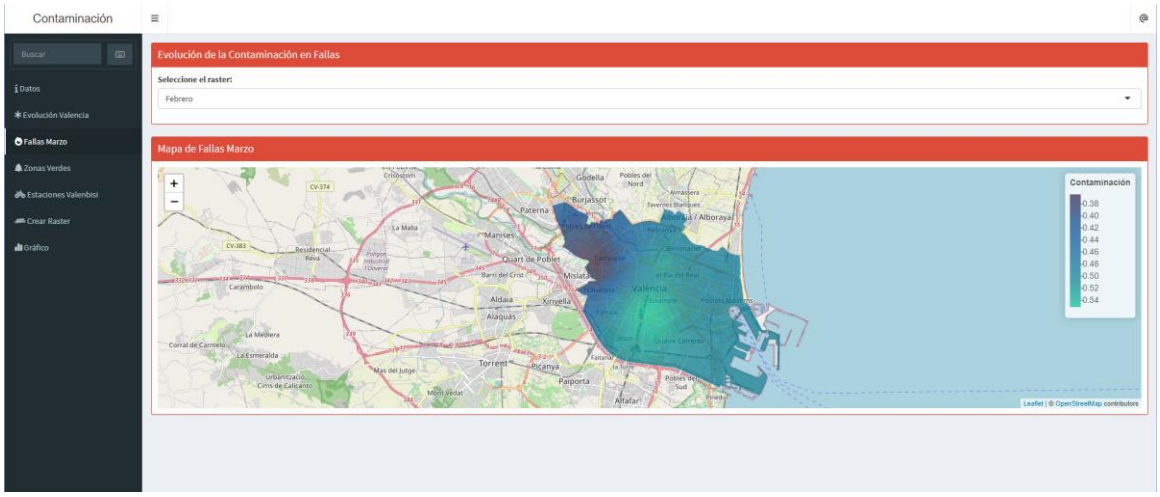
Evolución contaminación por año:



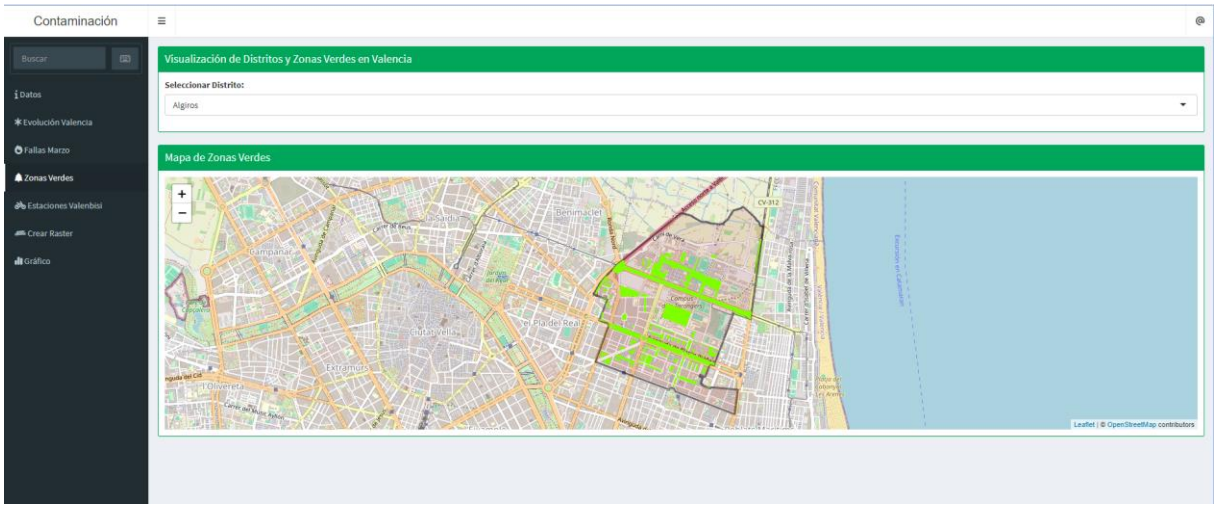
Se plantea una barra deslizante para seleccionar el año que quieres observar, además, se puede ver con un marcador las estaciones que proporcionan información para ese año (si pasas el ratón por encima se podrá ver el nombre).

Fallas y Zonas Verdes:

En fallas planteamos un caja con las opciones que tenemos, donde el usuario, puede elegir cualquiera de esta y visualizar el resultado.

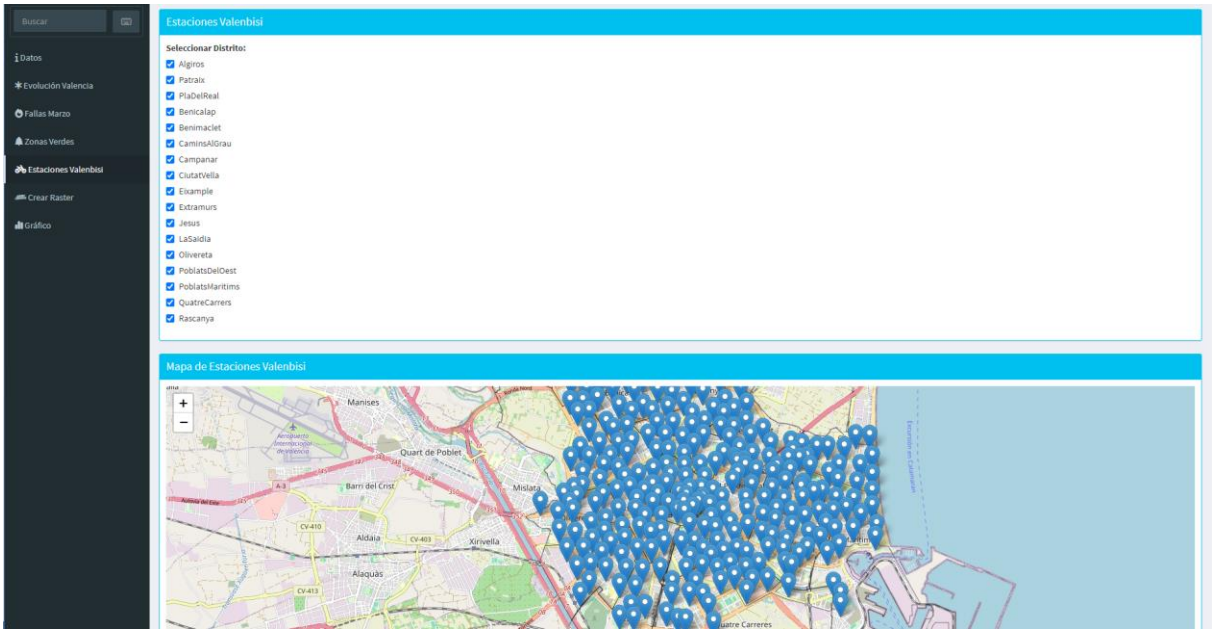


Lo mismo hacemos con la Zonas Verdes:



Valenbisi:

Por otro lado, en el gráfico de bicis, utilizamos un checkbox, para seleccionar o deseleccionar los distritos, en los cuales se mostraran las paradas que hay en ellos:



2.6. Diseño del cuadro de mandos

Explicar los elementos que conforman el cuadro de mandos.

2.7. Implementación

Detallar la implementación que se ha realizado en R.

Esta parte del código explicará la creación de la pestaña en la cual se pueden seleccionar los distritos que se quieren mostrar y en los seleccionadas se marcará la ubicación de estaciones de bici.

El código será explicado a continuación por partes, comenzando por definir el UI :

En esta parte del código creamos la caja de selección gracias a la función checkboxGroupInput(), donde sus parámetros son:

- Opciones: Todos los distritos, junto a su título “Seleccionar Distrito:”
- Selected: Los distritos que estarán seleccionados de manera predeterminada

```
# Tab para Estaciones Valenbisi
tabItem(
  tabName = "bicis_tab",
  fluidRow(
    box(
      title = "Estaciones Valenbisi",
      status = "info",
      solidHeader = TRUE,
      width = 12,
      checkboxGroupInput(
        "distrito_bicis",
        "Seleccionar Distrito:",
        choices = c(
          "Algiros",
          "Patraix",
          "PlaDelReal",
          "Benicalap",
          "Benimaclet",
          "CaminsAlGrau",
          "Campanar",
          "CiutatVella",
          "Eixample",
          "Extramurs",
          "Jesus",
          "LaSaidia",
          "Olivereta",
          "PoblatsDelOest",
          "PoblatsMaritims",
          "QuatreCarrers",
          "Rascanya"
        ),
        selected = c(
          "Algiros",
          "Patraix",
          "PlaDelReal",
          "Benicalap",
          "Benimaclet",
          "CaminsAlGrau",
          "Campanar",
          "CiutatVella",
          "Eixample",
          "Extramurs",
          "Jesus",
          "LaSaidia",
          "Olivereta",
          "PoblatsDelOest",
          "PoblatsMaritims",
          "QuatreCarrers",
          "Rascanya"
        )
      )
```

```
  ),
  fluidRow(
    box(
      title = "Mapa de Estaciones Valenbisi",
      status = "info",
      solidHeader = TRUE,
      width = 12,
      leafletOutput("map_bicis")
    )
  )
),
```

Por otra parte cargaremos un mapa mediante leafletOutput()

En cuanto a la definición del servidor observamos el siguiente código:

```r
output$map_bicis <- renderLeaflet({
  # Inicializar lista para almacenar los datos de los distritos seleccionados
  distrito_list <- list()
  puntos_list <- list()

  # Comprobar si se seleccionaron distritos
  if (length(input$distrito_bicis) > 0) {
    # Iterar sobre cada distrito seleccionado
    for (distrito_name in input$distrito_bicis) {
      # Construir la ruta del archivo shapefile para el distrito actual
      distrito_file <- paste0("distritos/", distrito_name, ".shp")
      puntos_file <-
        paste0("Bicis/estaciones/",
               "Estaciones",
               distrito_name,
               ".shp")

      # Leer los datos del distrito actual
      distrito_data <- st_read(distrito_file)
      distrito_data <- st_transform(distrito_data, crs = 4326)

      # Leer los datos de los puntos verdes del distrito actual
      puntos_data <- st_read(puntos_file)
      puntos_data <- st_transform(puntos_data, crs = 4326)

      # Almacenar los datos del distrito actual en la lista
      distrito_list[[distrito_name]] <- distrito_data
      puntos_list[[distrito_name]] <- puntos_data
    }
  }

  # Crear un mapa Leaflet y agregar polígonos para cada distrito seleccionado
  map <- leaflet() %>%
    addProviderTiles(provider = providers$OpenStreetMap) %>%
    setView(lng = -0.3763,
            lat = 39.4699,
            zoom = 12.5)
```

```r
    # Iterar sobre cada distrito seleccionado y agregar polígonos al mapa
    for (distrito_name in names(distrito_list)) {
      map <-
        addPolygons(
          map,
          data = distrito_list[[distrito_name]],
          fillOpacity = 0,
          color = "black",
          weight = 1,
          group = distrito_name
        )



      # Agregar capa de puntos de zonas verdes al mapa con marcadores
      map <- addMarkers(
        map,
        data = puntos_list[[distrito_name]],
        icon = NULL,
        # Usar el icono predeterminado
        popup = NULL,
        # Sin popups por defecto
        clusterOptions = NULL
      )  # No agrupar marcadores

    }

    # Retornar el mapa
    return(map)

  })
```

Inicialización de Variables:

Se definen dos listas vacías, distrito_list y puntos_list, que servirán para almacenar los datos de los distritos seleccionados y los puntos de interés (Estaciones de bici) respectivamente.

Verificación de Selecciones del Usuario:

Se comprueba si el usuario ha seleccionado algún distrito en la interfaz de usuario. Si hay selecciones, el código procede a cargar los datos correspondientes de esos distritos desde archivos shapefile.

Creación del Mapa:

Se genera un mapa interactivo Leaflet. Este mapa se establece inicialmente con una vista centrada en una ubicación específica y un nivel de zoom predeterminado.

Agregación de Polígonos de Distritos:

Para cada distrito seleccionado por el usuario, se agregan polígonos al mapa para representar los límites de esos distritos. Cada polígono se dibuja con un color de relleno azul transparente y un borde negro.

Agregación de Marcadores de estaciones de bici:

Para cada distrito seleccionado, se añaden marcadores circulares al mapa para representar la ubicación de los puntos de interés dentro de esos distritos. Estos marcadores se representan en verde y tienen un tamaño proporcional al nivel de zoom actual del mapa.

Finalización y Presentación del Mapa:

Una vez que se han agregado todos los polígonos y marcadores al mapa, se devuelve el mapa completo. Este mapa será renderizado en la interfaz de usuario de la aplicación web, proporcionando una visualización interactiva de los distritos y los puntos de interés seleccionados por el usuario.

Queremos tener también una opción para ver rasters de interpolación para fechas específicas. Para hacer esto hay que calcular los rasters de contaminación en R, convirtiendo, primero, los datos a un csv.

Hemos hecho este código en Python, utilizando las librerías pandas y geopandas:

```python
import geopandas as gpd
import pandas as pd


3 usages
def shp_to_df(filepath):
    gdf = gpd.read_file(filepath)
    gdf = gdf.to_crs(epsg=4326)

    gdf['x'] = gdf.geometry.x
    gdf['y'] = gdf.geometry.y
    return gdf


filepaths = [
    '/Users/erandoni/Downloads/Miniproyecto 2/Miniproyecto/SUBSETS_N_ESTACIONES/2010_2018_6_NOT_NULL.shp',
    '/Users/erandoni/Downloads/Miniproyecto 2/Miniproyecto/SUBSETS_N_ESTACIONES/2018_2021_7_NOT_NULL.shp',
    '/Users/erandoni/Downloads/Miniproyecto 2/Miniproyecto/SUBSETS_N_ESTACIONES/2022_2022_8_NOT_NULL.shp'
]

data = shp_to_df(filepaths[0])
data2 = shp_to_df(filepaths[1])
data3 = shp_to_df(filepaths[2])
data3['ICA'] = data3['IMC']
data = pd.concat([data, data2, data3])
data.to_csv('2010_2022_merged.csv', index=False)
```

Nos fijamos que además de convertir a csv, también creamos una columna para los valores de X e Y. Estos, hay que pasarlos a un CRS de 4326.

Ahora podemos empezar la interpolación.

Primero cargamos nuestros datos, y también los shapefiles de los distritos de Valencia que vamos a usar más adelante.

```
data <- read.csv('/Users/erandoni/PycharmProjects/FARS/2010_2022_merged.csv', header=T)
data <- data[!is.na(data$ICA), ]
shapefile <- st_read("/Users/erandoni/Downloads/reproj84_districtos.shp")
```

De nuestros datos, obtenemos las coordenadas, y los límites para la interpolación. Ya que, los datos contienen las estaciones atmosféricas, hay que usar un offset de 0.1 para incluir el resto de la ciudad. También convertimos la columna de fecha a tipo DateTime, y creamos una columna con el año.

```
x_min <- min(data$x)-0.1
x_max <- max(data$x)+0.1
y_min <- min(data$y)-0.1
y_max <- max(data$y)+0.1
data$f_buena <- as.Date(data$f_buena)
data$year <- year(data$f_buena)
```

Para hacer la interpolación necesitamos crear un grid, al cual vamos a aplicar la fórmula del IDW. Para esto usamos nuestros límites de X e Y, y elegimos una resolución de 500 por 500 para el equilibrio entre velocidad y precisión.

```
grid_points <- expand.grid(x = seq(x_min, x_max, length.out = n_points_x),
                           y = seq(y_min, y_max, length.out = n_points_y))
coordinates(grid_points) = ~ x + y
gridded(grid_points) <- TRUE
fullgrid(grid_points) <- TRUE
```

Después creamos la UI, que tiene opciones para elegir un año y un rango de fechas para la interpolación, y además un output del mapa y un botón para hacer el plot.

```r
ui <- fluidPage(
  titlePanel("Date Range Selector"),

  sidebarLayout(
    sidebarPanel(
      selectInput("year", "Select Year:", choices = 2010:2022),
      dateRangeInput("dateRange", "Select Date Range:",
                     start = "2010-01-01", end = "2022-12-31", format = "mm-dd"),
      actionButton("plotButton", "Plot")
    ),

    mainPanel(
      leafletOutput("map")
    )
  )
)
```

En el server utilizamos un observeEvent para esperar que el usuario toque el botón y posteriormente mostrar el plot, y así obtenemos los datos del subset que el usuario ha elegido.

```r
server <- function(input, output) {
  # Reactive expression for selected data
  observeEvent(input$plotButton, {
    year <- input$year
    dateRange <- as.Date(input$dateRange)
    start_date <- as.Date(paste(year, format(dateRange[1], "%m-%d"), sep = "-"))
    end_date <- as.Date(paste(year, format(dateRange[2], "%m-%d"), sep = "-"))
    data_subset <- subset(data, f_buena >= start_date & f_buena <= end_date)
```

Definimos una paleta de colores para el mapa, con límites definidos para consistencia entre grafos de tiempos diferentes:

```r
output$map <- renderLeaflet({
  # Definir la paleta estilo Turbo
  turbo_palette <- colorRampPalette(c("#301E43", "#02468E", "#0360A0", "#028090", "#02C39A", "#87D68D", "#F0E874", "#E76F51", "#CF3C38", "#960018"))
  turbo_palette(...)
  # Crear una secuencia de colores
  turbo_colors <- turbo_palette(100)

  # Definir los límites de la paleta (ajusta estos valores según tu necesidad)
  palette_min <- 0.2
  palette_max <- 1

  # Función para mapear los valores del raster a la paleta de colores
  pal <- colorNumeric(
    palette = turbo_colors,
    na.color = "transparent",
    domain = c(palette_min, palette_max)
  )
```

Ahora estamos listos para la interpolación.

Usamos la función de idw, y convertimos el resultante a un raster. Hacemos un crop y un mask con el shapefile que cargamos de distritos, y añadimos el raster para mostrarlo encima del mapa de OpenStreetMap, centrado en la ciudad de Valencia.

```
idw_res <- idw(ICA ~ 1, locations = data_subset, newdata = grid_points, idp = 2)
raster_idw <- raster(idw_res)
projection(raster_idw) <- CRS(crs(shapefile))
raster_idw <- crop(raster_idw,shapefile)
raster_idw <- mask(raster_idw, shapefile)
leaflet() %>%
  addProviderTiles(provider = providers$OpenStreetMap) %>%
  addRasterImage(raster_idw, colors = pal, opacity = 0.7) %>%
  setView(lng = -0.3763, lat = 39.4699, zoom = 12)  %>%
  addLegend(pal = pal,
            values = values(raster_idw),
            title = "Contaminación",
            opacity = 0.7)
```

Aquí podemos observar el subset predefinido cuando abrimos la aplicación:

Para finalizar la implantación, una vez tenemos lista toda la información que queremos mostrar, utilizamos la librería “shinydashboard”, para unir todas estas en una misma aplicación con un entorno gráfico más agradable.

Primero, definimos la ui con el dashboardPage(), definiendo dentro de este, el dashboardHeader(), el dashboardSidebar() y el dashboardBody().

- dashboardHeader(): En este definimos el título que queremos darle a la cabecera y añadimos un botón que muestra el correo de los participantes del trabajo.

```
ui <- dashboardPage(title = 'Dashboard', skin = 'black',
                    dashboardHeader(

                      title = "Contaminación",
                      tags$li(actionLink("openModal", label = "",
                                         icon = icon("at")),
                              class = "dropdown")
```

- dashboardSidebar(): En este apartado, definimos el menú con las items que podemos elegir:

```
dashboardSidebar(
  sidebarSearchForm("searchText","buttonSearch","Buscar",icon = shiny::icon("keyboard")),
  sidebarMenu(
    menuItem("Datos", tabName = "introduccion_tab",icon = icon("info")),
    menuItem("Evolución Valencia", tabName = "valencia_tab", icon = icon("asterisk")),
    menuItem("Fallas Marzo", tabName = "fallas_tab", icon = icon("fire")),
    menuItem("Zonas Verdes", tabName = "verdes_tab",icon = icon("tree")),
    menuItem("Estaciones Valenbisi", tabName = "bicis_tab",icon = icon("bicycle")),
    menuItem("Crear Raster", tabName = "creaRaster_tab",icon = icon("angrycreative")),
    menuItem("Gráfico", tabName = "grafico_tab",icon = icon("chart-simple"))
  )
```

- dashboardBody(): En este apartado, añadimos todos los items, definiéndolos cada uno en un tabItem y añadiendo características visuales para la aplicación, como por ejemplo añadimos un status al box, dependiendo del item que estemos seleccionando. En todos estos, usamos el solidHeader = TRUE, lo que significa que las cabecera de todas las box que usamos está opaca, ya que si fuera solidHeader = FALSE, la cabecera sería transparente. Por otro lado, añadimos un width de 12, lo que hace que las box estén más extendidas, adaptándose más a la anchura de la pantalla. Un ejemplo de un tabItem sería el siguiente:

```
dashboardBody(
  tabItems(
    # Tab para la Introducción
    tabItem(
      tabName = "introduccion_tab",
      fluidRow(
        box(
          title = "Información de los datos",
          status = "primary",
          solidHeader = TRUE,
          width = 12,
          uiOutput("texto_estatico")
        )
      )
    ),
```

Llamamos a las funciones glo_evo y glo_fallas, que nos calculan el máximo y mínimo de las capas raster, de esta manera todas las representaciones tienen la misma escala.

```
glo_evo <- unlist(globales_evolucion())
global_max <- glo_evo[1]
global_min <- glo_evo[2]


glo_fallas <- unlist(globales_fallas())
fallas_max <- glo_fallas[1]
fallas_min <- glo_fallas[2]
```

Después de definir ui y llamado a las funciones, procedemos a definir el server

En este, definimos tantos output como items tenemos para mostrar, como por ejemplo el del texto_estatico del ejemplo anterior (en el dashboardBody):

```
output$texto_estatico <- renderUI({
  div(
    class = "texto-estatico",
    p("- 2.5PM: Son partículas con un tamaño de 2,5 micras de diámetro, según los expertos.
   Se encuentran directamente en la atmósfera, como resultado de fenómenos naturales
   (incendios forestales o emisiones volcánicas) o de actividades humanas (actividades
   agrícolas o de construcción, polvo en suspensión, actividades industriales, etc.)."),
    p("- Óxidos de Nitrógeno (NO,NO2,NOx): Los óxidos de nitrógeno son una mezcla de gases
   compuestos de nitrógeno y oxígeno. El monóxido de nitrógeno y el dióxido de nitrógeno
   constituyen dos de los óxidos de nitrógeno más importantes toxicológicamente hablando.
   Los óxidos de nitrógeno son liberados al aire desde el escape de vehículos motorizados,
   de la combustión de todo tipo."),
    p("- O3: El ozono se considera un contaminante ambiental,
   ya que a elevadas concentraciones puede provocar daños en la salud como irritar el
   sistema respiratorio, agravar el asma y las enfermedades pulmonares crónicas,
   reducir la función pulmonar, disminuir la esperanza de vida."),
    p("- ICA: El ICA viene definido por la siguiente formula:
  0.4  (Promedio diario NO2/50)+ 0.3 (Promedio diario PM 2.5/25)+0.3(Promedio diario de O3/120).
  El conjunto de nuestros gases contaminantes nos ayudará a tener una mejor percepción de la contaminación que tenemos.")
  )
})
```

En este también definimos el botón de los participantes que hemos creado en la ui:

```
observeEvent(input$openModal, {
  showModal(
    modalDialog(title = "Participantes",
                p("Contacto:"),
                p("damaco@alumni.uv.es"),
                p("palodo2@alumni.uv.es"),
                p("juango9@alumni.uv.es"),
                p("vicarno@alumni.uv.es"),
                p("edo7@alumni.uv.es")
                )
  )
})
```

Toda esta aplicación ha sido subida a shinyApp, el problema que nos ha sucedido, es que al ejecutar el programa, demanda mucha RAM, por lo que en el plan gratis de esta plataforma, no llega a funcionar correctamente.

A la hora de la ejecución del programa, solo funciona correctamente en un ordenador con características medio-altas.

LINK: https://pablopezdom.shinyapps.io/Shiny_miniproyecto/

3. Resultados

Presentación de los datos obtenidos de manera clara y organizada.

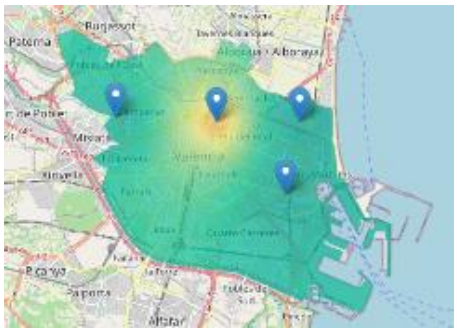
Utilización de tablas, gráficos u otros recursos visuales para facilitar la comprensión.

Análisis de los resultados, resaltando las tendencias o patrones observados.

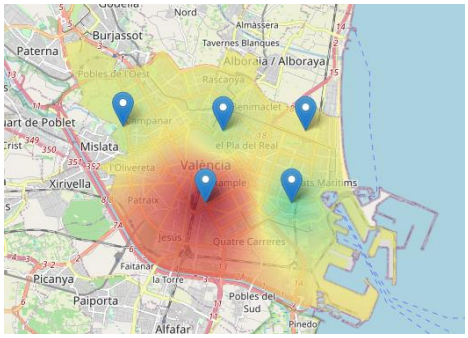
Evolución de la contaminación:

Como una primera toma de contacto, se puede observar la evolución del ICA, que a partir de 2019 disminuye significativamente. Podría tener relación con la cuarentena de la pandemia de Covid-19, ya que las fechas coinciden.

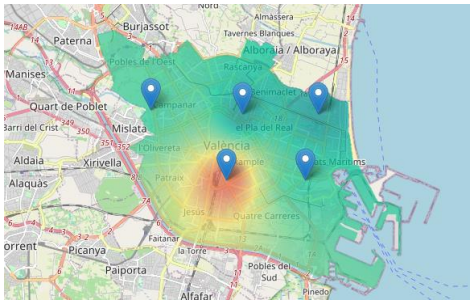
2013:



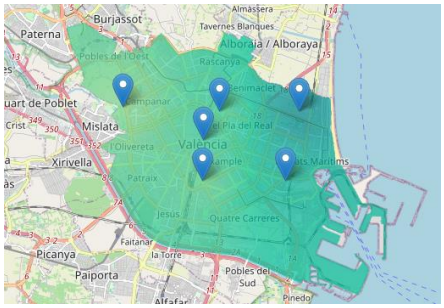
2015:



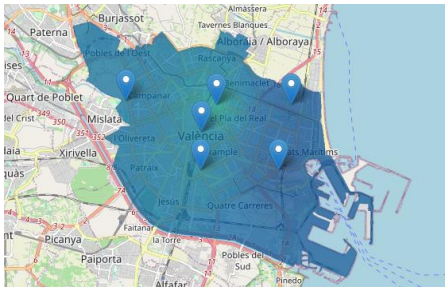
2018:



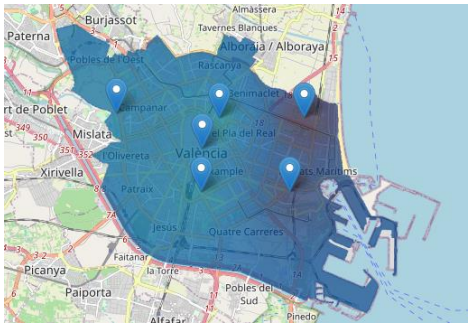
2019:



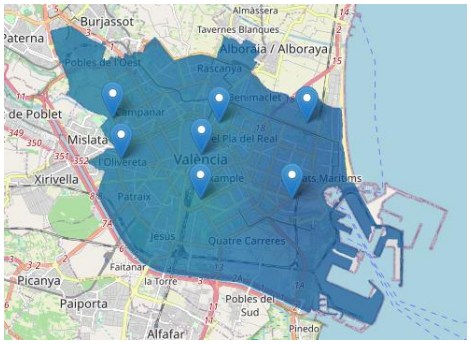
2020:



2021:

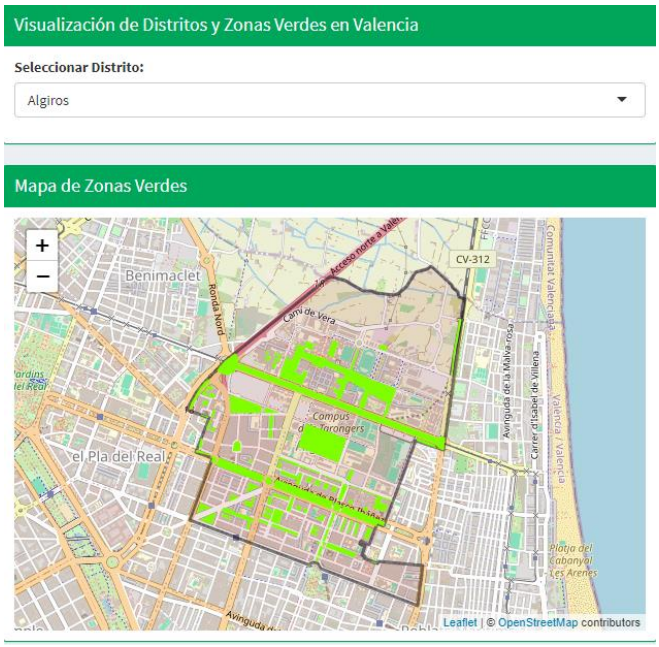


2022:



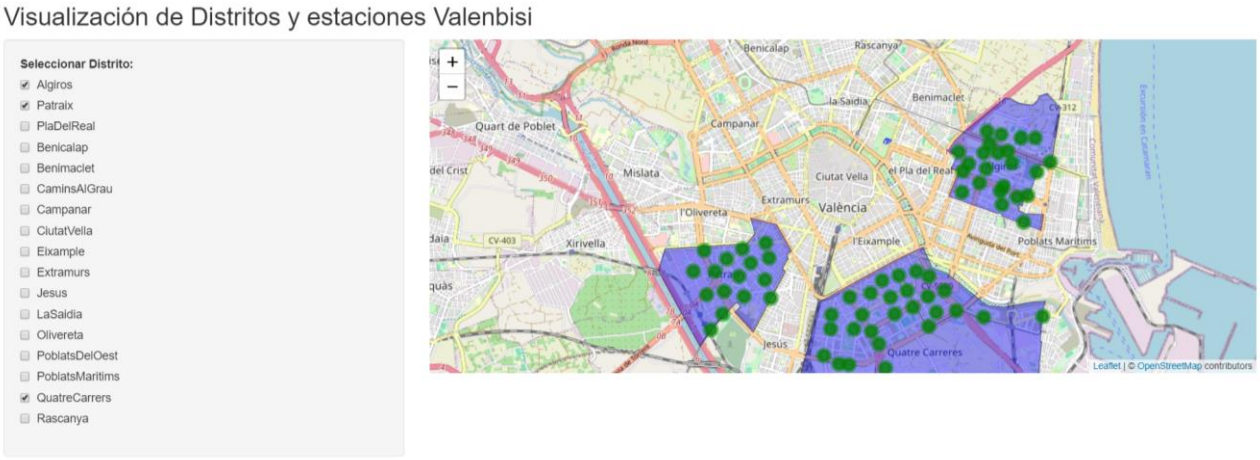
Zonas Verdes:

En esta podemos seleccionar el distrito que queremos observar, y el programa nos muestra las zonas verdes que existen en este.



Estaciones valenbisi:

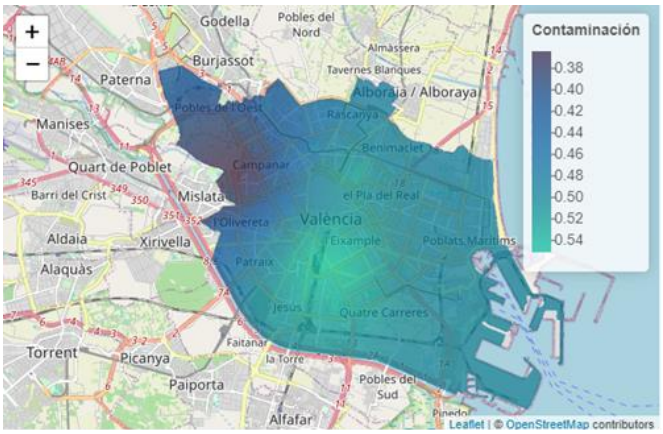
Podemos observar como en un distrito como Patraix, lugar donde en el resto de mapas se observa mayor contaminación, hay un número menor de estaciones a nivel general. Por ello podemos relacionar que a un número mayor de estaciones de bici será menor el nivel de contaminación gracias a la sostenibilidad de este servicio.



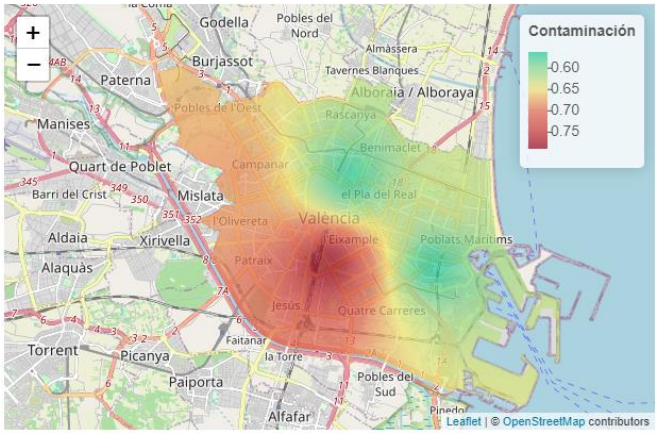
Fallas:

En cuanto a nuestro análisis sobre las fallas, vemos la evolución que tiene contaminación en estas fechas:

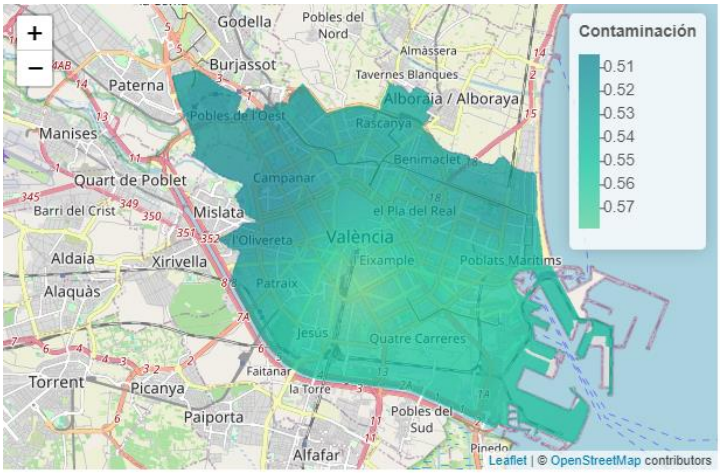
- Febrero (antes de fallas):



- Marzo (justo en las semana de fallas):



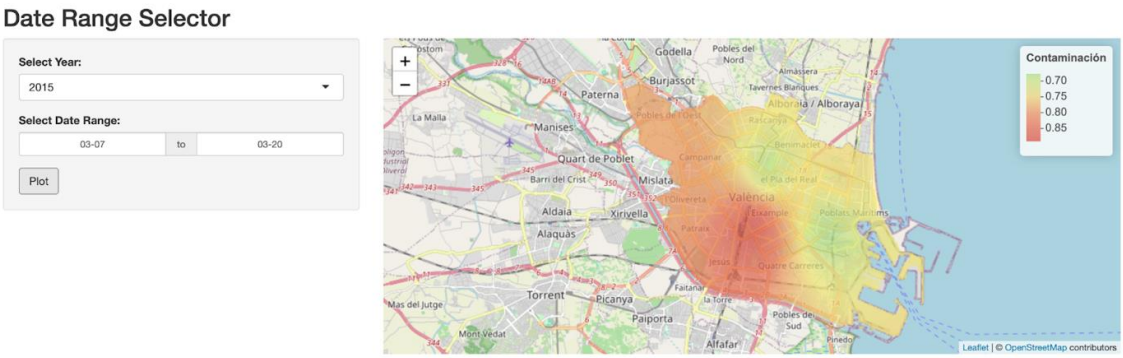
- Marzo (finales de marzo, una vez ya han cavado las fallas):



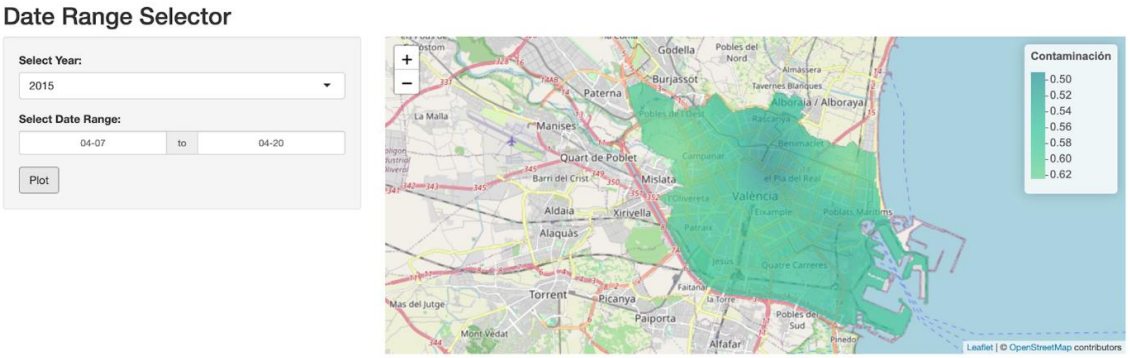
Creador de Raster:

En este podemos visualizar un subset entre dos fechas en un año determinado:

Por ejemplo, tenemos este gráfico que muestra la contaminación en la semana de fallas:



Otro ejemplo, donde mostramos los mismos días pero en el mes de abril del mismo año:



4. Discusión

Los resultados obtenidos han sido los esperados relativamente, ya que no esperábamos que la evolución de la contaminación desde 2020 hasta 2022, fuera tan buena.

Contemplábamos que para el año 2019-2020 (pandemia), la contaminación disminuyera, pero lo que no esperábamos, esa tendencia se quedara durante un par de años más.

Otra de las razones del estudio ha sido ver el impacto de las fallas en la calidad del aire de Valencia. En este nos hemos encontrado lo esperado, ya que la contaminación antes de esta celebración es mucho más baja que al finalizar dicha fiesta. Y luego, si observamos esta durante la misma semana de fallas, el índice de calidad del aire aumenta drásticamente (cuanto más alto este índice, peor).

La calidad de los datos ha sido un problema presente durante todo el estudio, ya que hemos tenido que descartar muchos registros debido a datos faltantes en estos. Esa ha sido la mayor dificultad que nos hemos encontrado al realizar este trabajo.

5. Conclusiones

Los hallazgos más significativos han sido:

La reducción notable de la contaminación en la ciudad de Valencia después de la pandemia ha sido uno de los hallazgos más significativos en términos de contaminación. Esto puede deberse a la disminución de la actividad industrial, comercial y de transporte durante los períodos de confinamiento y restricciones. Con menos vehículos en las calles, se generaron menos emisiones de gases de escape y partículas contaminantes, por lo que se obtuvo una mejora significativa de la calidad del aire. Esto no solo benefició a los residentes locales, sino también al medio ambiente en general, reduciendo la huella de carbono y reduciendo el cambio climático.

El aumento de la contaminación durante la semana de las Fallas en Valencia ha sido otro hallazgo importante. Durante esta semana se produce un fenómeno peculiar que afecta negativamente la calidad del aire, reflejado en un empeoramiento del Índice de Calidad del Aire (ICA). Durante esta semana, la ciudad se llena de turistas y se llevan a cabo numerosos espectáculos pirotécnicos, lo que se suma al uso desmesurado de pólvora y a la quema de cientos de fallas que emiten humo durante horas. Este fenómeno no solo afecta la calidad del aire durante la semana de fallas, sino que también puede tener repercusiones a largo plazo en la salud de los habitantes. Si bien las celebraciones son una parte importante de la cultura local y atraen a turistas de todo el mundo, también es crucial abordar los impactos ambientales y tomar medidas para reducir la contaminación atmosférica asociada con estas festividades.

Una forma de contribuir a la mejoría del ICA, sería usar transporte público (por eso en nuestra aplicación se sitúan las estaciones de “ValenBisi”), aumentar la proporción de zonas verdes o una distribución mejor de ellas.

A nivel de la salud personal, se recomienda el uso de zonas verdes para pasar el tiempo, en vez de pasarlo por zonas más urbanizadas. Ya que los municipios con más zonas verdes tienen un nivel de contaminantes más bajo.

6. Referencias

Realizamos una llamada al servicio 012, el cual centraliza el suministro de información administrativa de la Generalitat en un único número de teléfono 012 (o 963 866 000) o a través de un servicio de consulta online. El cual nos ayudó con los límites permitidos de las diferentes partículas contaminantes en la ciudad de Valencia.