

Tarea 3

June 22, 2025

1 Tarea 3

Tópicos Selectos De Matemáticas Aplicadas II: Análisis de Datos con Python Fecha de entrega: Domingo 22 de junio

Alan Badillo Salas

1. Guarde en un Dataframe los datos de los precios de las acciones de Tesla (*TSLA.csv* en Github) y realice lo siguiente:

- Obtenga el precio de cierre más alto y el precio de cierre más bajo a partir del 2020.
- Obtenga las gráficas de las series de tiempo de cada año utilizando subplots.
- Obtenga las gráficas de caja por mes de los precios de cierre del 2022.
- ¿Los precios de cierre de todo el 2022 se distribuyen normalmente? Justifique su respuesta.

```
[99]: import pandas
```

```
tesla = pandas.read_csv("tesla.csv", index_col=0, parse_dates=True)
```

```
tesla.head()
```

```
[99]:
```

	Open	High	Low	Close	Adj Close
Date					
2020-01-02	28.299999	28.713333	28.114000	28.684000	28.684000
2020-01-03	29.366667	30.266666	29.128000	29.534000	29.534000
2020-01-06	29.364668	30.104000	29.333332	30.102667	30.102667
2020-01-07	30.760000	31.441999	30.224001	31.270666	31.270666
2020-01-08	31.580000	33.232666	31.215334	32.809334	32.809334

```
[100]: tesla.sort_values(by="Close", ascending=False).head(1)
```

```
[100]:
```

	Open	High	Low	Close	Adj Close
Date					
2021-11-04	411.470001	414.496674	405.666656	409.970001	409.970001

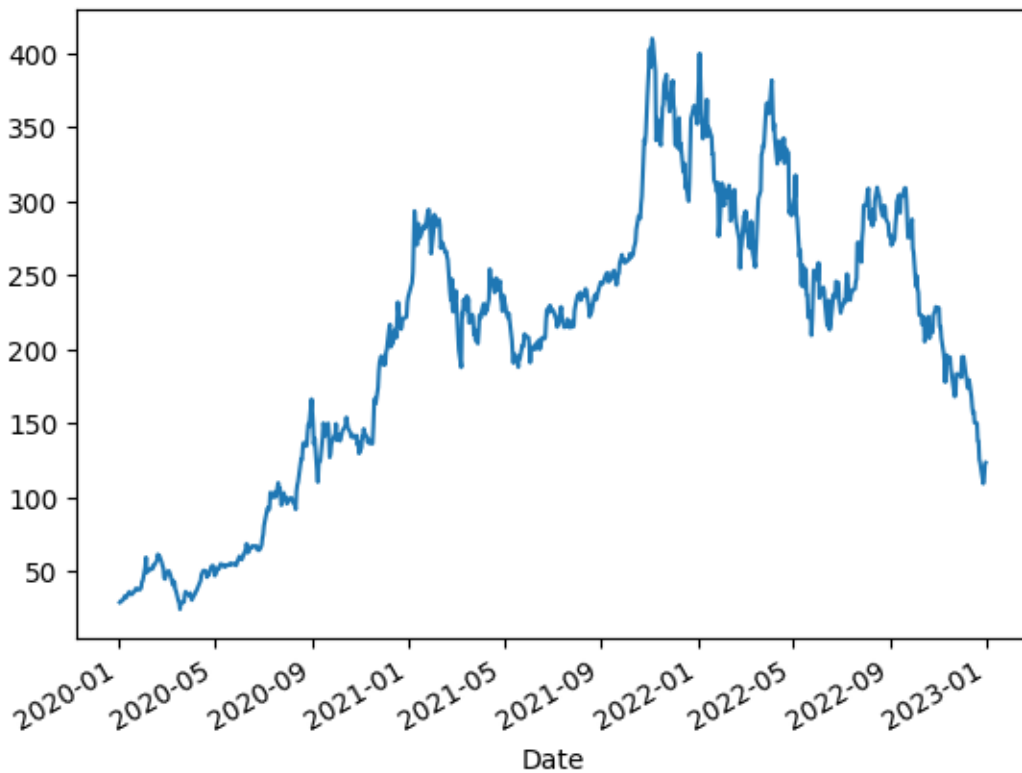
```
[101]: tesla.sort_values(by="Close", ascending=True).head(1)
```

```
[101]:
```

	Open	High	Low	Close	Adj Close
Date					
2020-03-18	25.933332	26.990667	23.367332	24.081333	24.081333

```
[102]: tesla["Close"].plot()
```

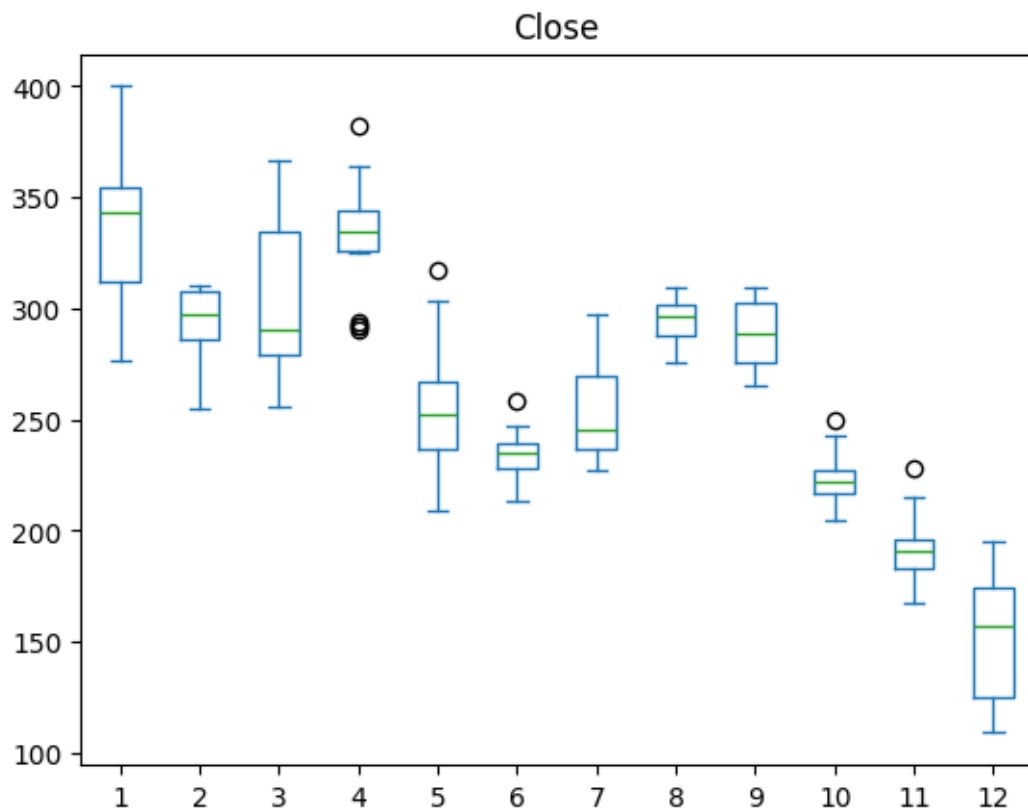
```
[102]: <Axes: xlabel='Date'>
```



```
[103]: tesla["mes"] = tesla.index.map(lambda fecha: fecha.month)
tesla["anio"] = tesla.index.map(lambda fecha: fecha.year)

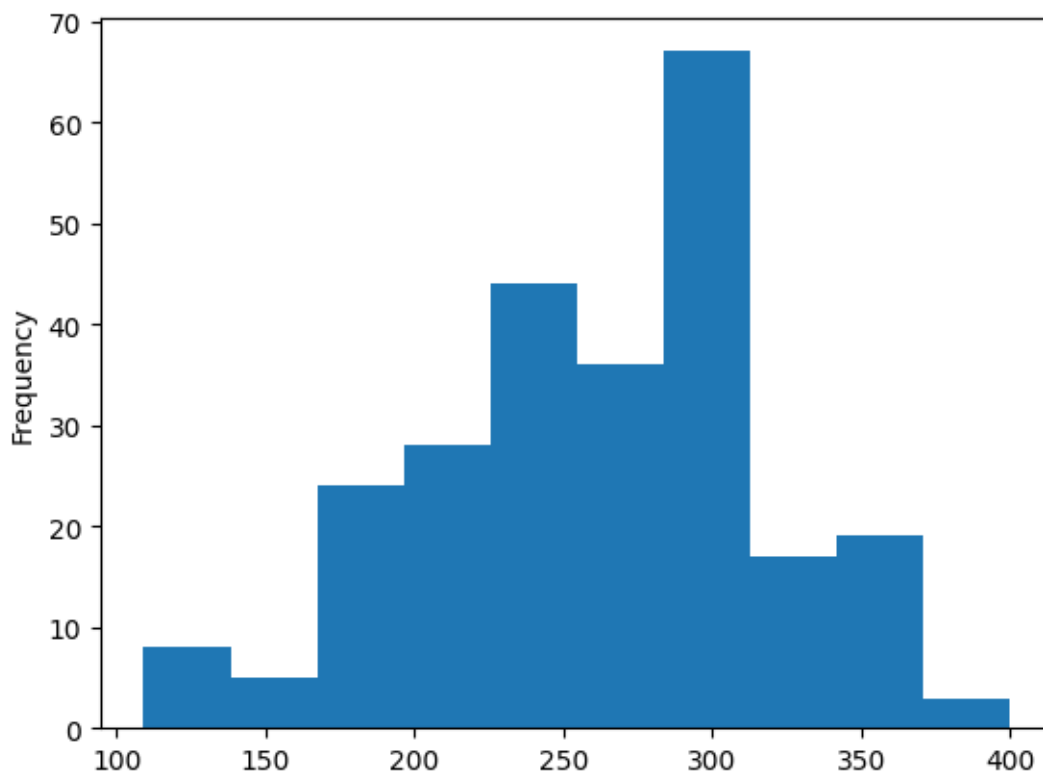
tesla[tesla["anio"] == 2022][["mes", "Close"]].plot.box("mes")
```

```
[103]: Close    Axes(0.125,0.11;0.775x0.77)
dtype: object
```



```
[104]: tesla[tesla["anio"] == 2022]["Close"].plot.hist()
```

```
[104]: <Axes: ylabel='Frequency'>
```



```
[105]: tesla[tesla["anio"] == 2022][["Close"]].skew()
```

```
[105]: Close    -0.307273
      dtype: float64
```

```
[106]: tesla[tesla["anio"] == 2022][["Close"]].kurt()
```

```
[106]: Close    -0.223269
      dtype: float64
```

Los precios del 2022 no parecen distribuirse de forma normal, tienen un ligero sesgo y curtosis negativa.

```
[107]: from scipy.stats import shapiro

r_value, p_value = shapiro(tesla[tesla["anio"] == 2022][["Close"]])

print(f"r-value: {r_value:.2f}")
print(f"p-value: {p_value:.2f}")
```

```
r-value: 0.99
p-value: 0.02
```

Vemos que el p-value es menor a 0.05 lo que sugiere rechazar la hipótesis nula H_0 de que los datos se distribuyen de forma normal.

2. Guarde en un Dataframe los datos de los contaminantes principales de la zona centro de la CDMX durante el año 2021 (*rama2021.xlsx* en Github) y realice lo siguiente:

- Obtenga las gráficas de series de tiempo de los contaminantes.
- Obtenga los histogramas de los contaminantes.
- Obtenga un Dataframe con el resumen de 5 números, asimetría y kurtosis para todos los contaminantes.
- Obtenga los gráficos de caja de los contaminantes. ¿Cuál contaminante fue el que más detectado durante 2021?
- Realice una prueba de contraste de normalidad de los datos.
- De acuerdo a la prueba obtenida anteriormente. ¿Cuál es el coeficiente de correlación apropiado para obtener la correlación entre los contaminantes?
- Obtenga una matriz de correlación de los contaminantes y un mapa de calor. Además, obtenga un Dataframe que muestre la correlación 2 a 2 entre los contaminantes.
- Obtenga un diagrama de dispersión de los 2 contaminantes donde existe más correlación.
- Obtenga una gráfica que muestre todos los diagrama de dispersión entre todos los contaminantes.
- Compruebe si existe correlación entre los contaminantes y los factores meteorológicos contenidos en *redmet_2021.csv* en Github.
- Escriba sus conclusiones.

```
[108]: import pandas

rama2021 = pandas.read_csv("rama_2021.csv", index_col=0, parse_dates=True)

rama2021.head()
```

```
[108]:
```

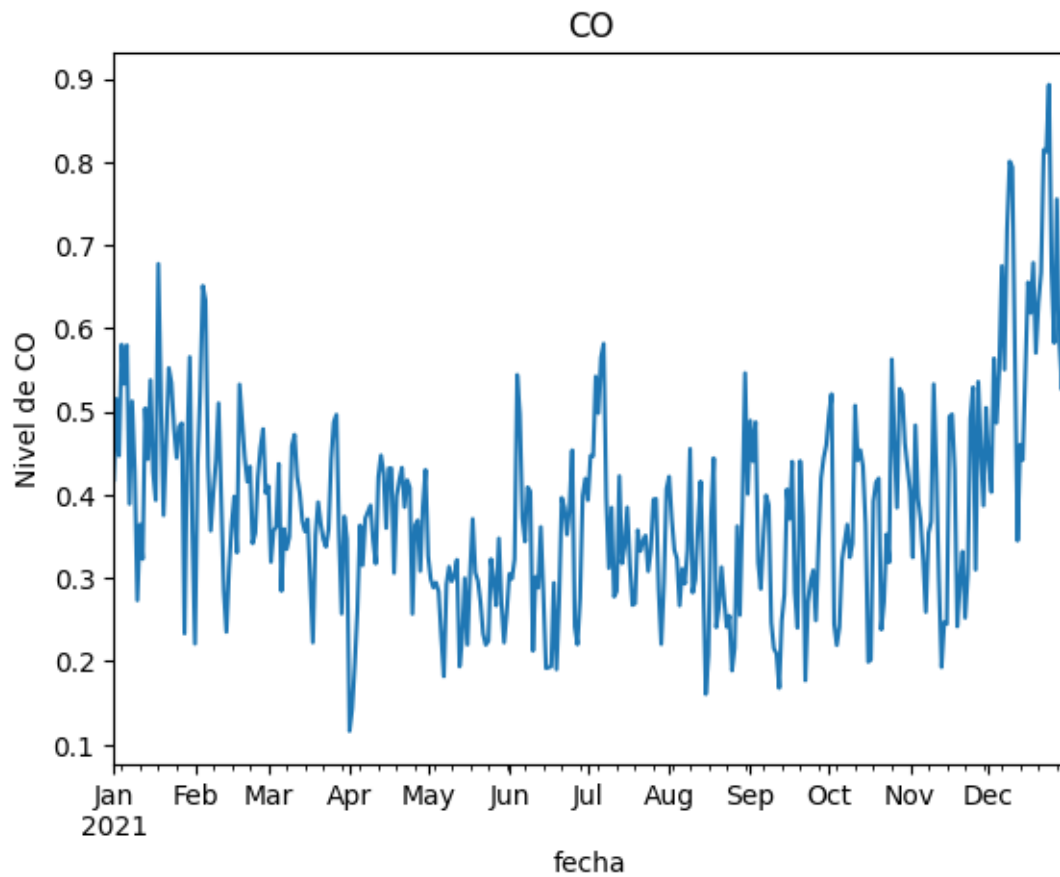
	CO	NO	NO2	NOX	O3	PM10 \
fecha						
2021-01-01	0.417782	5.712963	16.977273	23.261574	29.765494	54.327273
2021-01-02	0.515087	14.289894	26.782609	42.090426	33.647969	48.196481
2021-01-03	0.446905	12.328571	22.773256	35.714286	28.135823	39.091988
2021-01-04	0.580398	22.323040	31.306483	54.418052	28.824532	52.486239
2021-01-05	0.533105	19.386091	29.376218	49.995204	28.941374	55.916667

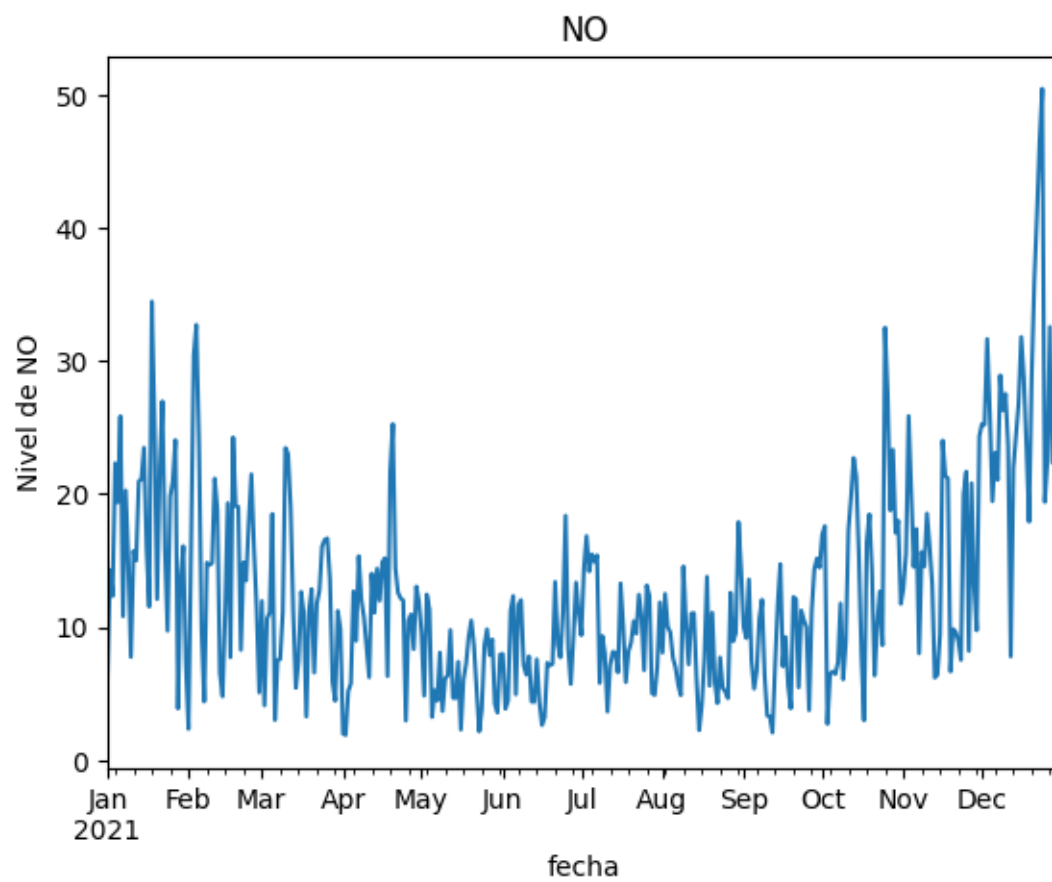
	PM25	S02
fecha		
2021-01-01	28.594855	1.125912
2021-01-02	23.465409	1.725000
2021-01-03	16.423676	1.548983

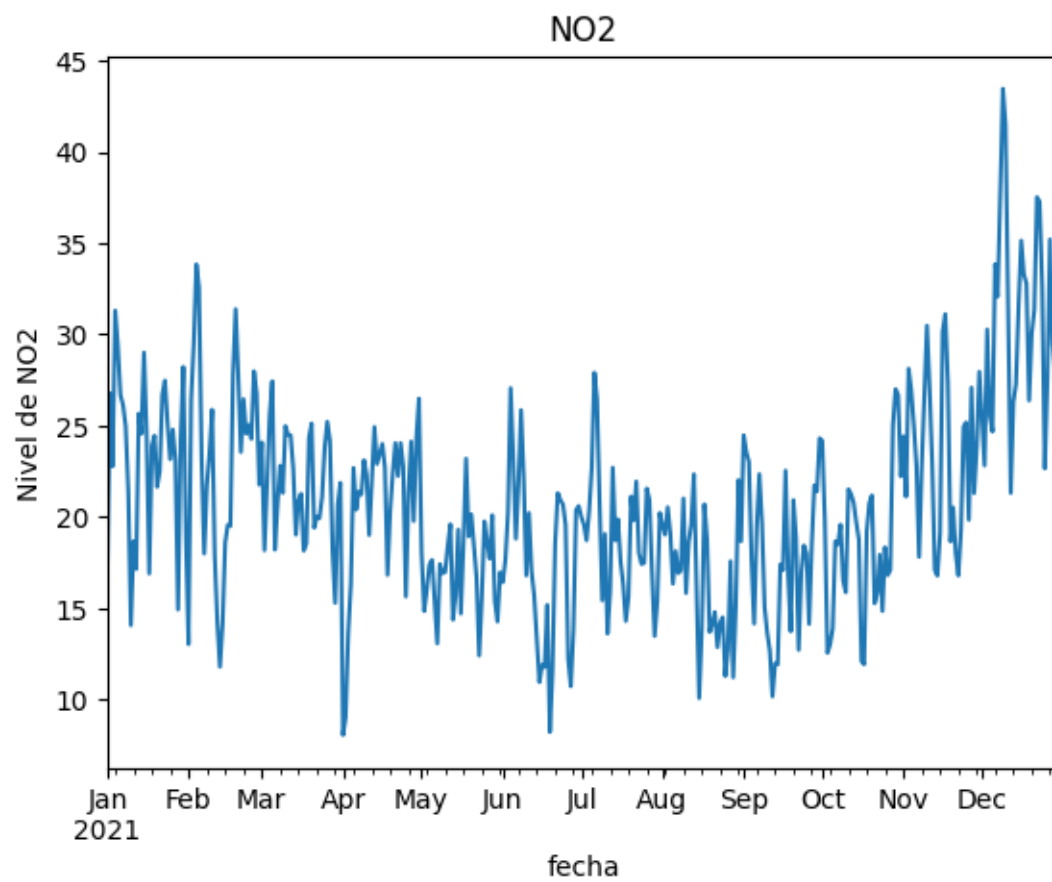
```
2021-01-04 23.160000 3.410019
2021-01-05 22.694444 5.251366
```

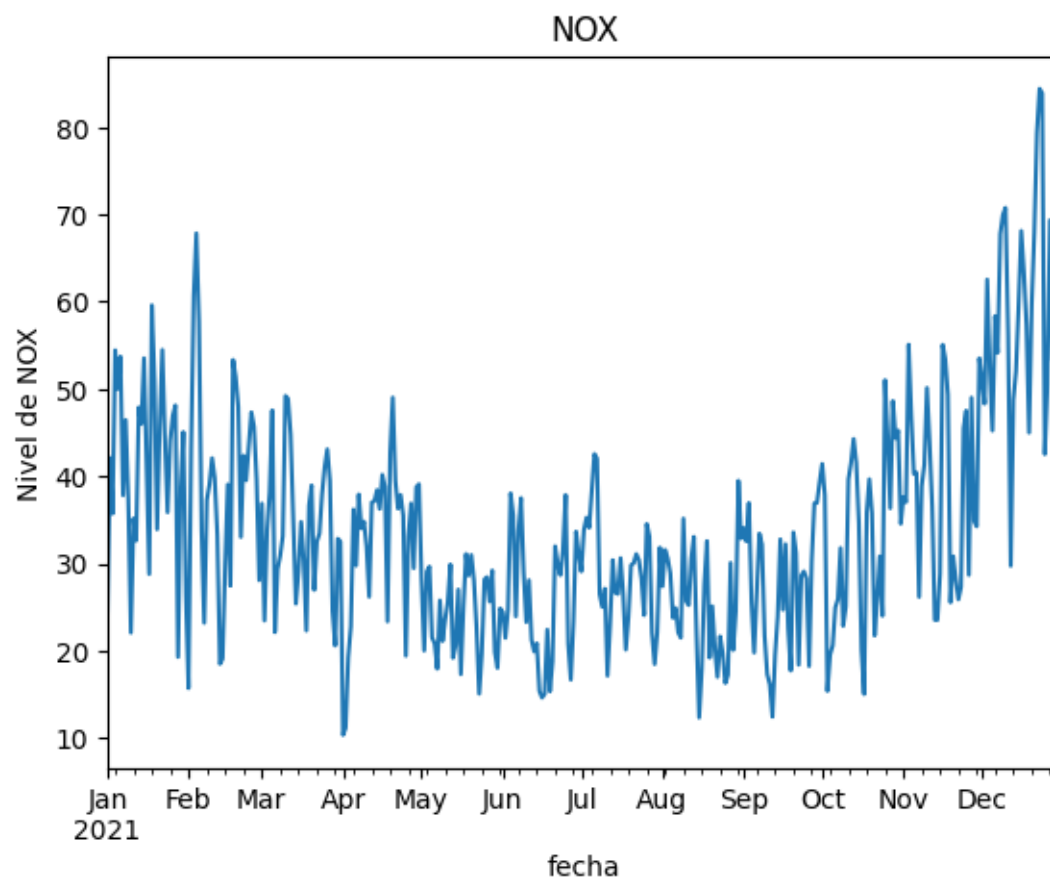
```
[109]: import matplotlib.pyplot as pyplot

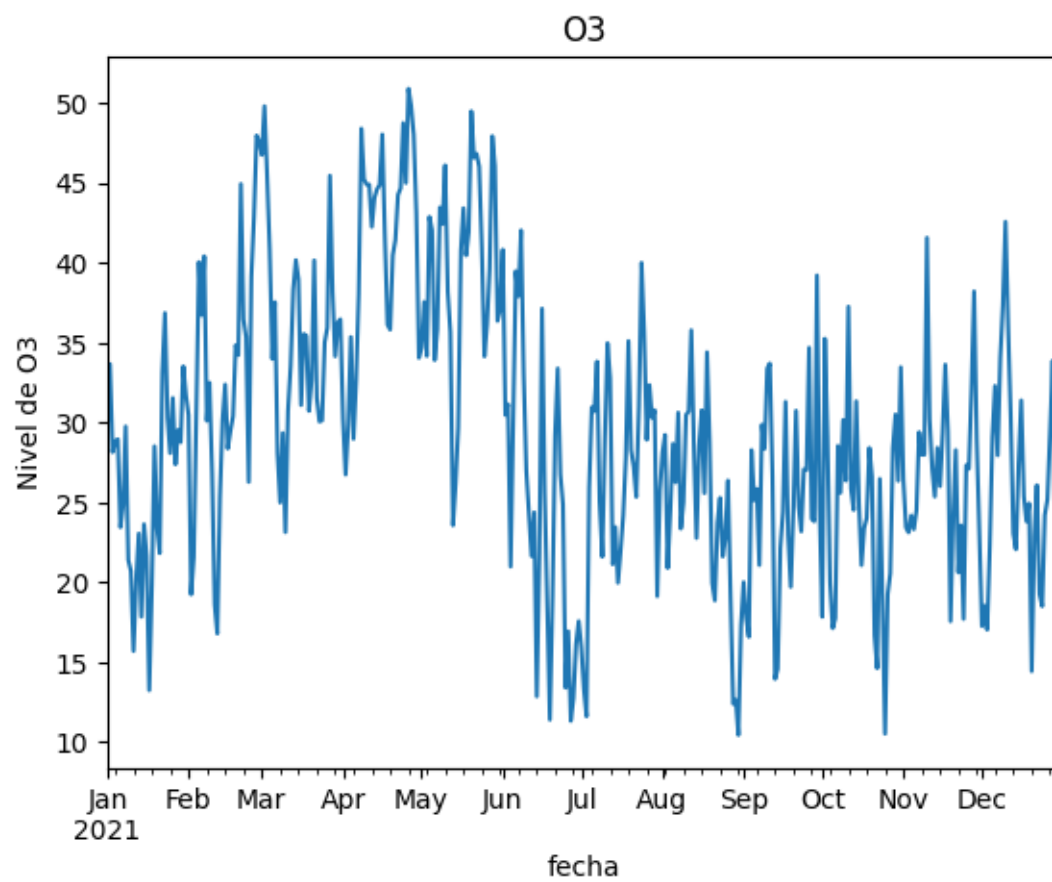
for column in rama2021.columns:
    rama2021[column].plot()
    pyplot.title(column)
    pyplot.ylabel(f"Nivel de {column}")
    pyplot.show()
```

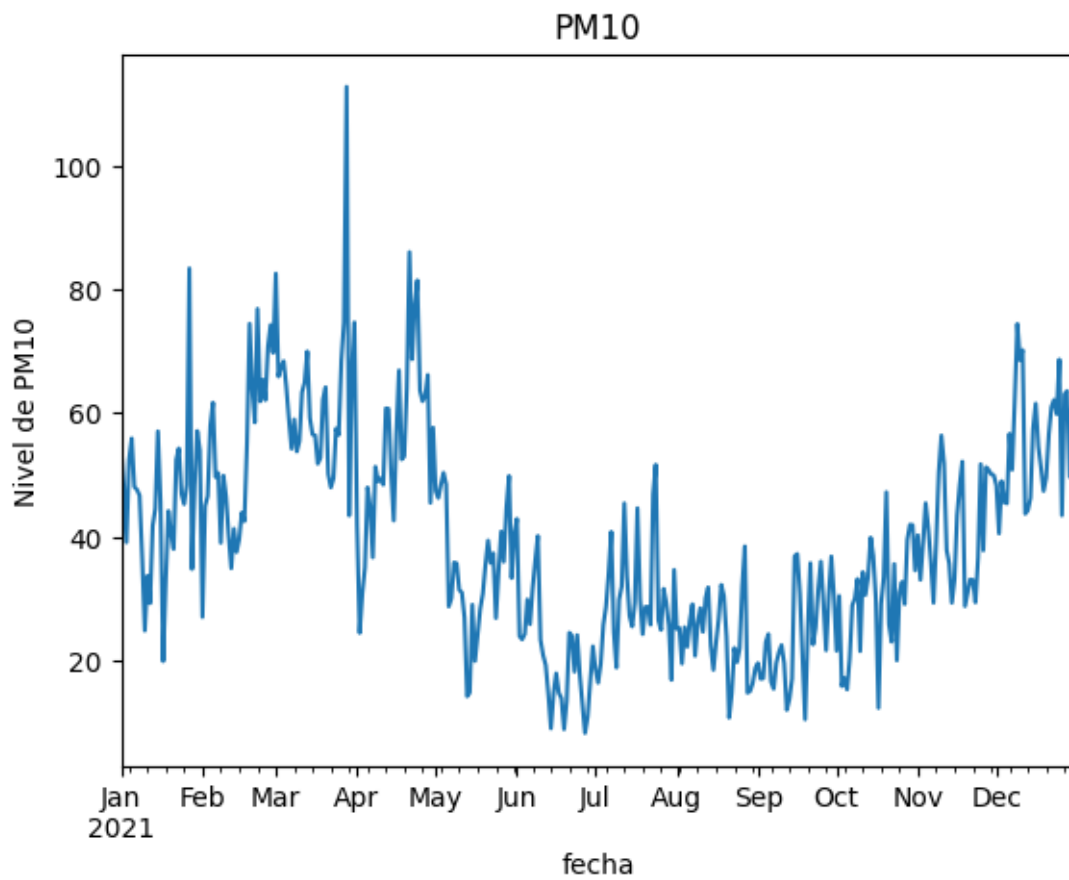


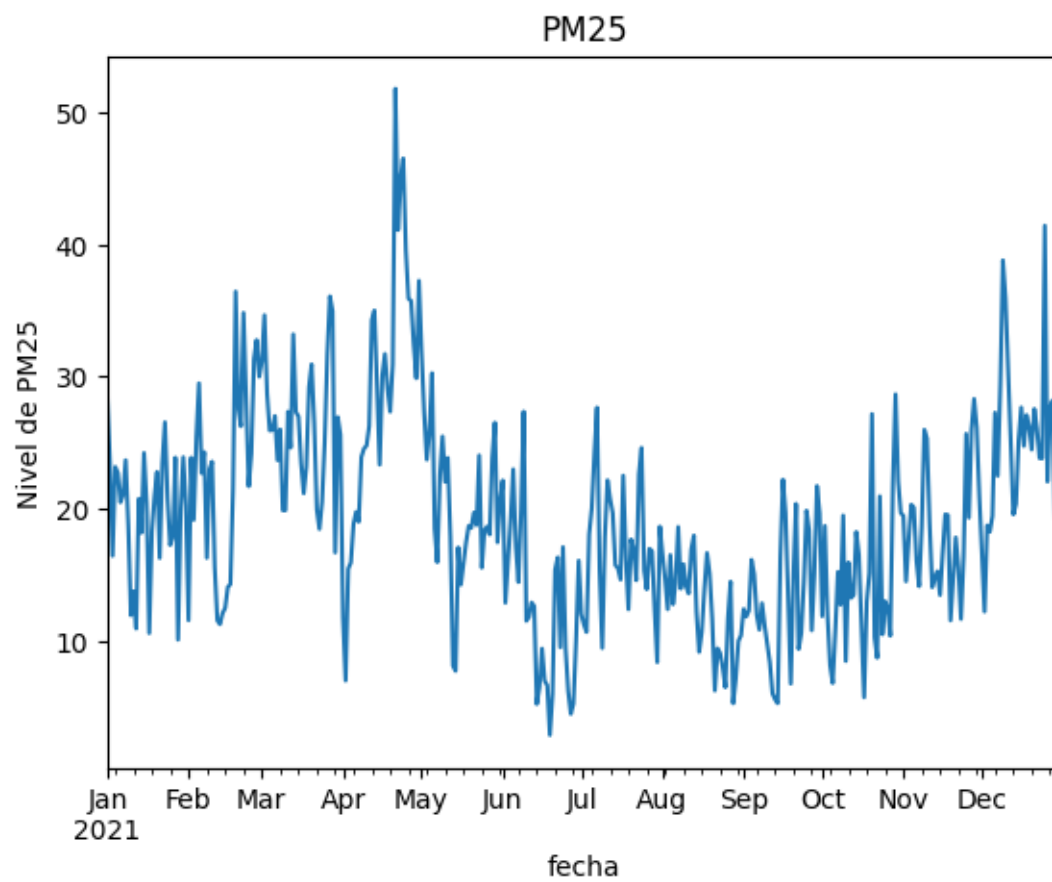


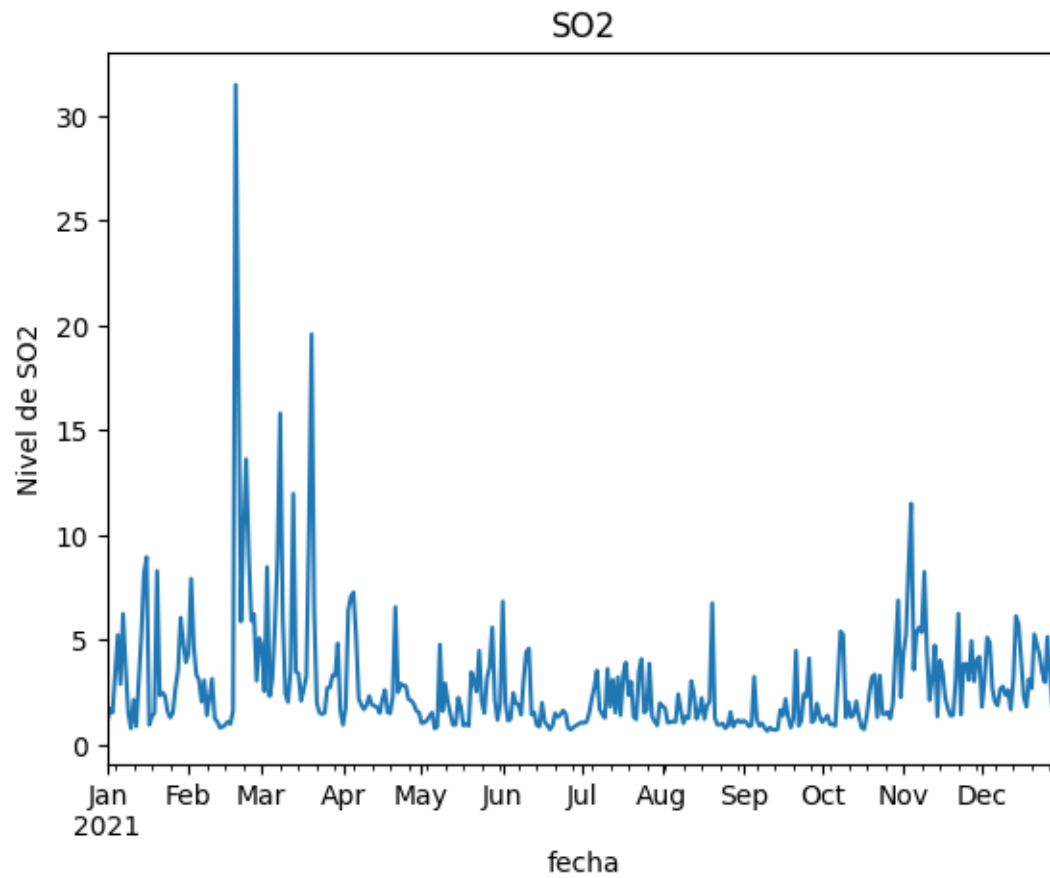




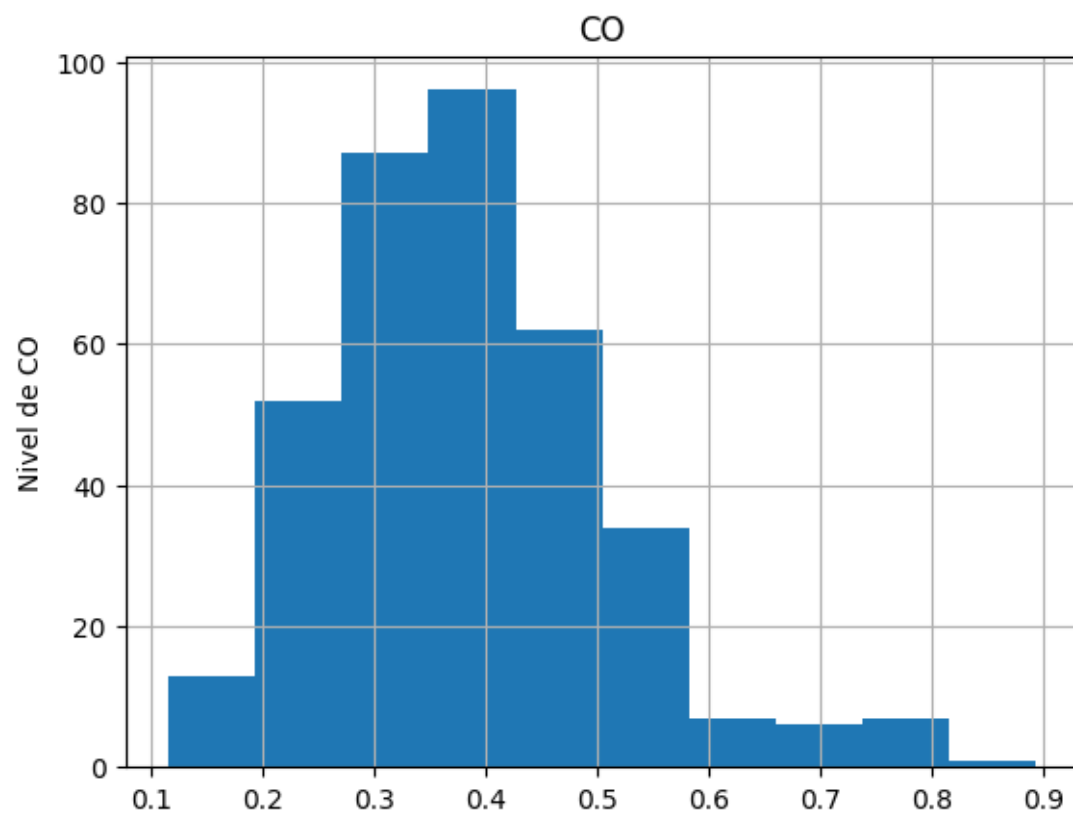


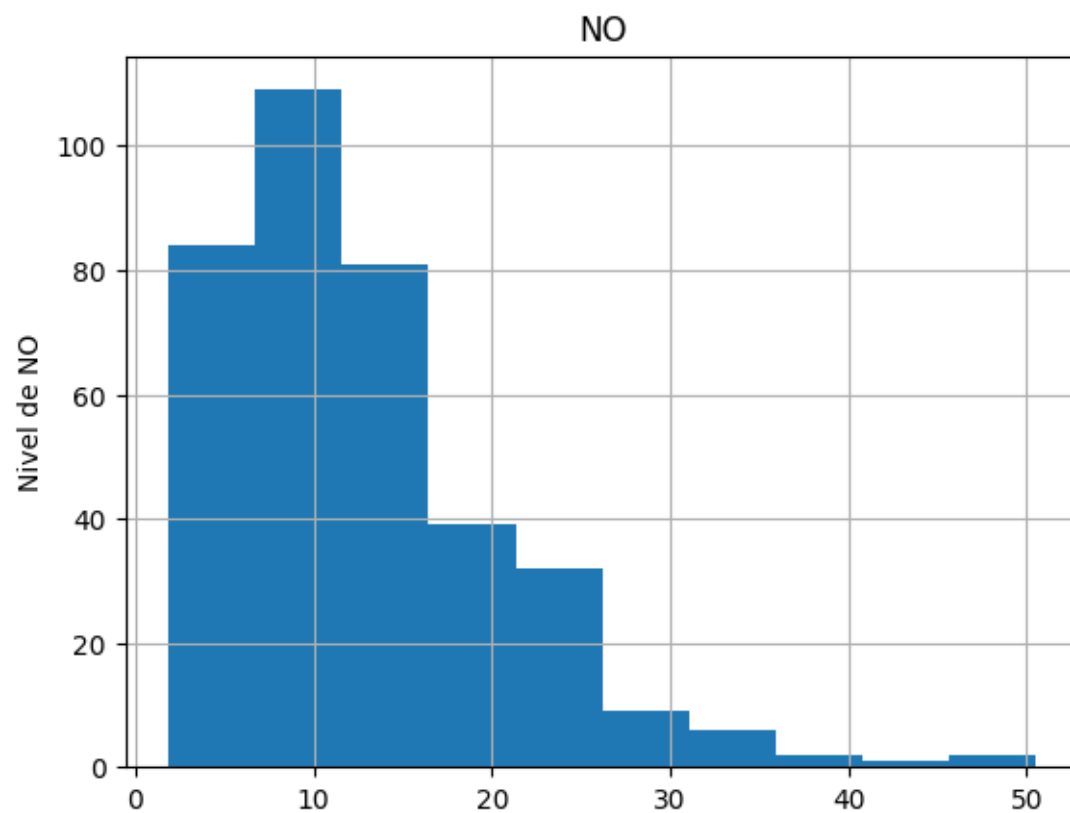


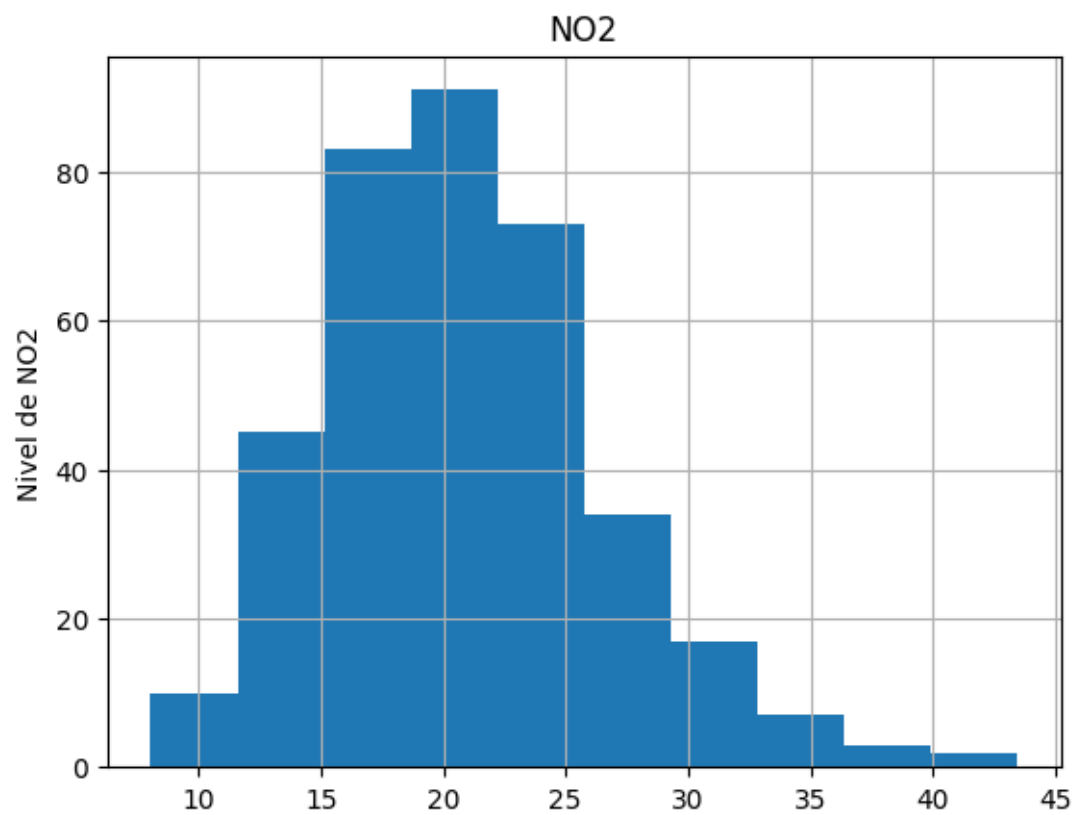


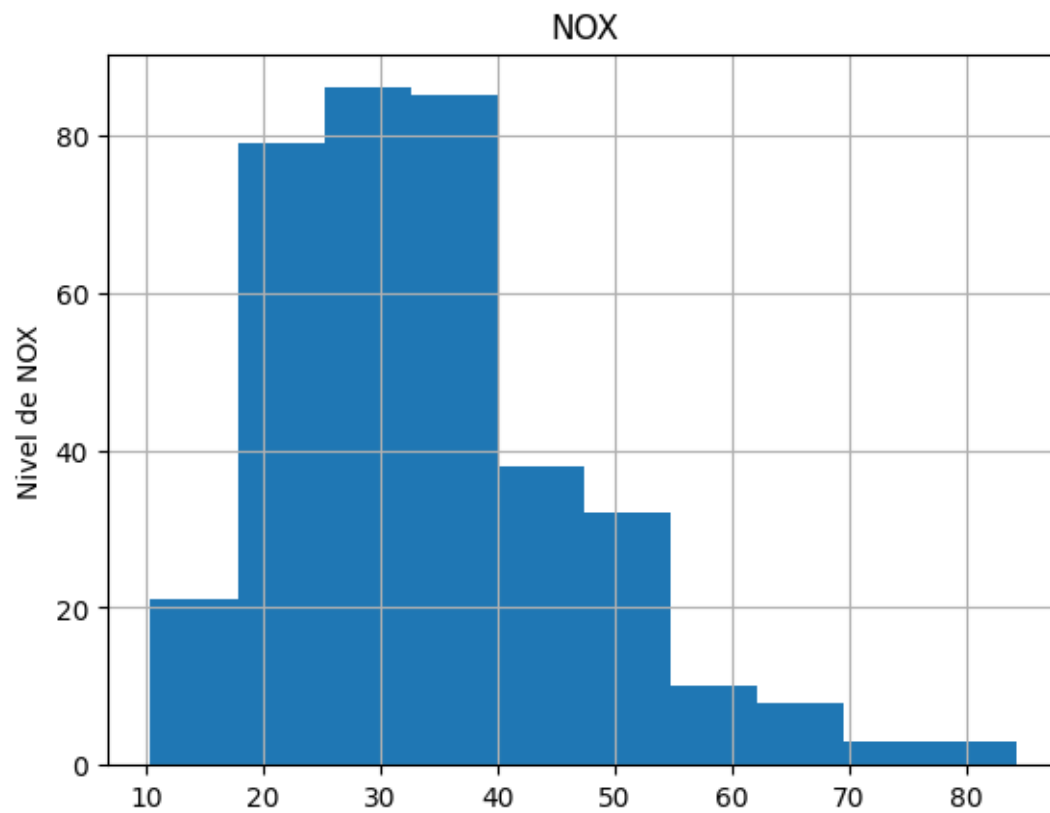


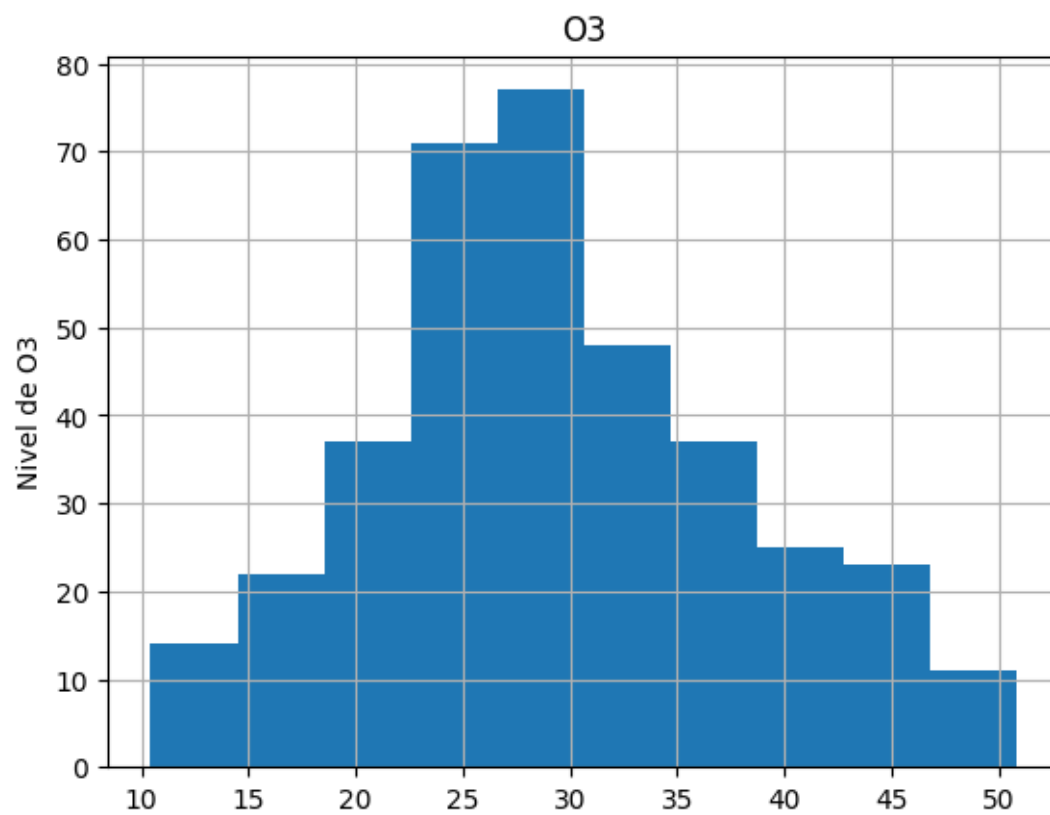
```
[110]: for column in rama2021.columns:
        rama2021[column].hist()
        pyplot.title(column)
        pyplot.ylabel(f"Nivel de {column}")
        pyplot.show()
```

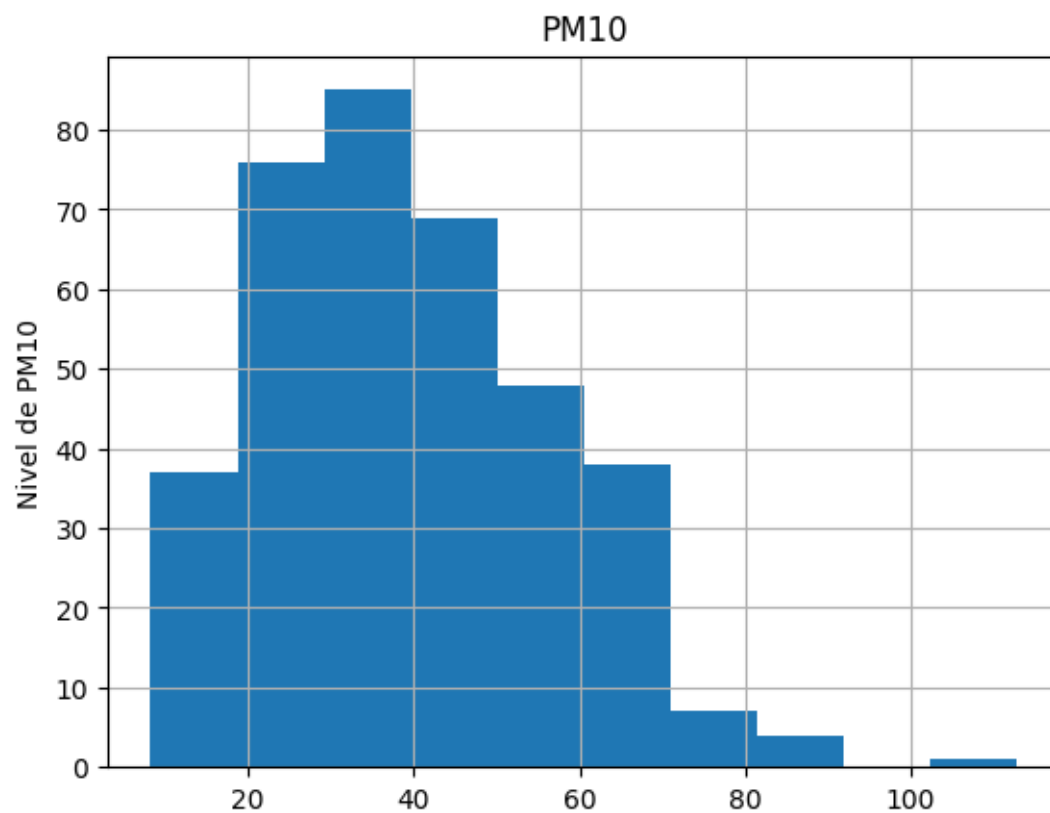


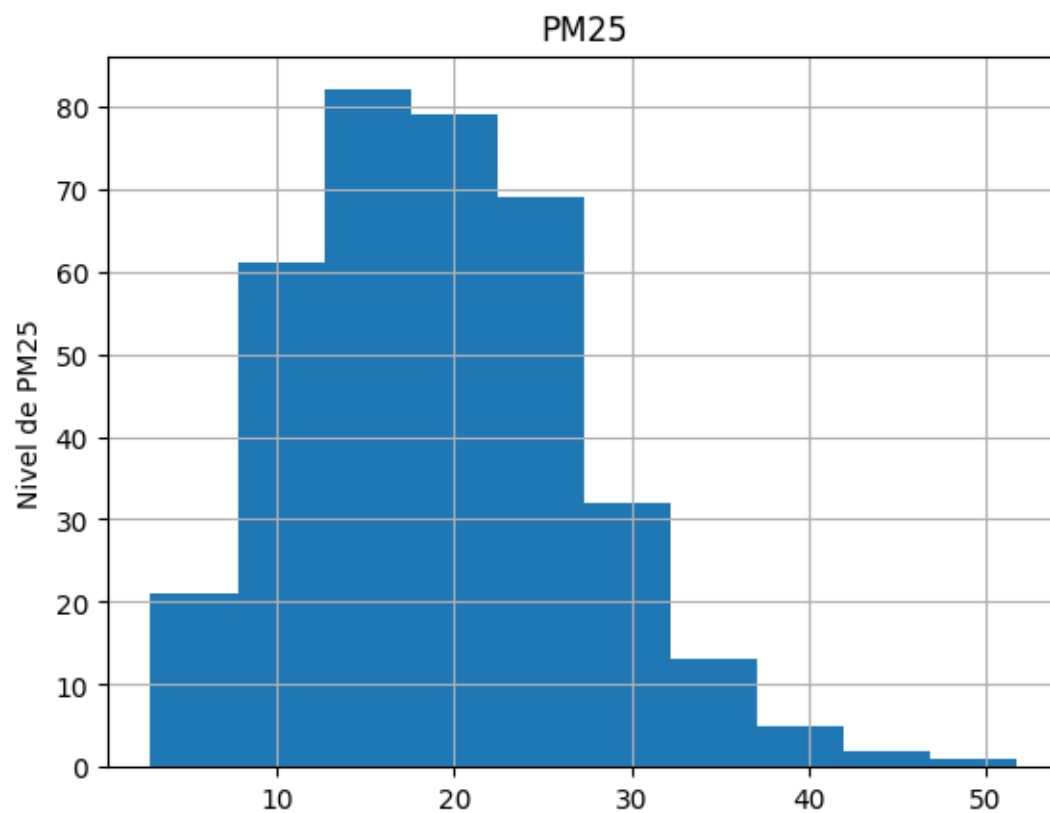


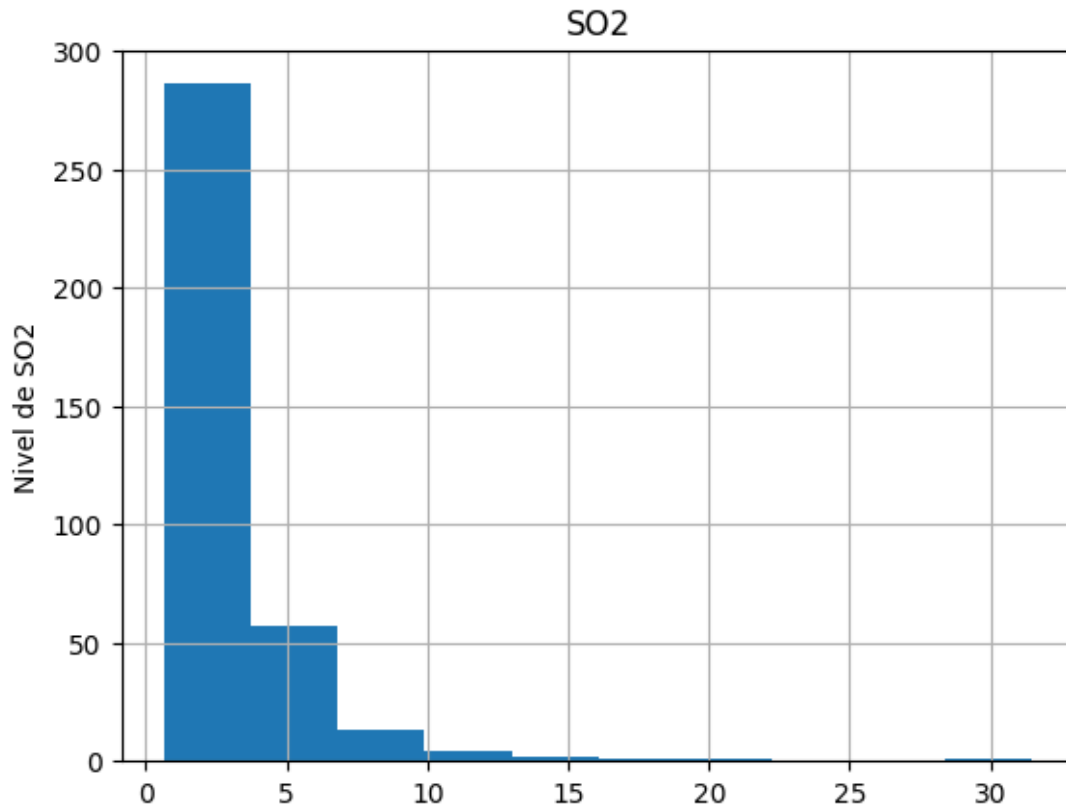












```
[111]: for column in rama2021.columns:
        print(f"Contaminante: {column}")
        print("-" * 30)
        print(rama2021[column].sample(5))
        print("-" * 30)
        print(f"Asimetría: {rama2021[column].skew()}")
        print(f"Curtosis: {rama2021[column].kurt()}")
        print("-" * 30)
        print()
```

Contaminante: CO

```
-----
fecha
2021-11-09    0.367570
2021-02-15    0.363219
2021-10-13    0.453606
2021-12-04    0.486253
2021-11-20    0.295441
Name: CO, dtype: float64
-----
```

Asimetría: 0.9803378359693984
Curtosis: 1.610452138988229

Contaminante: NO

fecha

2021-01-16	15.865429
2021-09-07	10.628176
2021-08-25	5.138889
2021-12-28	22.344749
2021-05-30	3.553435

Name: NO, dtype: float64

Asimetría: 1.3195268056823903
Curtosis: 2.470978105190348

Contaminante: NO2

fecha

2021-09-15	17.406957
2021-06-05	23.215000
2021-10-13	20.780936
2021-06-17	11.781250
2021-11-15	19.104882

Name: NO2, dtype: float64

Asimetría: 0.6677544286841419
Curtosis: 0.8894470857027419

Contaminante: NOX

fecha

2021-02-02	43.204360
2021-11-04	47.074468
2021-03-23	33.423611
2021-04-25	19.424242
2021-12-02	48.301527

Name: NOX, dtype: float64

Asimetría: 0.9862719607358327
Curtosis: 1.2238361549880197

Contaminante: O3

```

fecha
2021-12-20    14.418470
2021-07-21    27.237049
2021-11-29    28.964286
2021-10-26    19.222920
2021-02-01    30.461039
Name: 03, dtype: float64
-----
Asimetría: 0.2571565188875678
Curtosis:  -0.3631178743198409
-----

```

Contaminante: PM10

```

-----
fecha
2021-04-30    57.670554
2021-11-16    43.458050
2021-02-28    69.755224
2021-11-25    51.740139
2021-06-29    16.678571
Name: PM10, dtype: float64
-----
Asimetría: 0.5376491593982847
Curtosis:  0.11742207838921592
-----

```

Contaminante: PM25

```

-----
fecha
2021-11-17    19.561404
2021-08-06    14.359773
2021-06-10    11.495082
2021-01-15    24.212329
2021-11-23    11.635328
Name: PM25, dtype: float64
-----
Asimetría: 0.6357826591214513
Curtosis:  0.6865608860130203
-----

```

Contaminante: SO2

```

-----
fecha
2021-11-16    3.456674
2021-04-26    2.182292
2021-04-29    1.634000
2021-07-29    1.116412
2021-09-14    0.762340

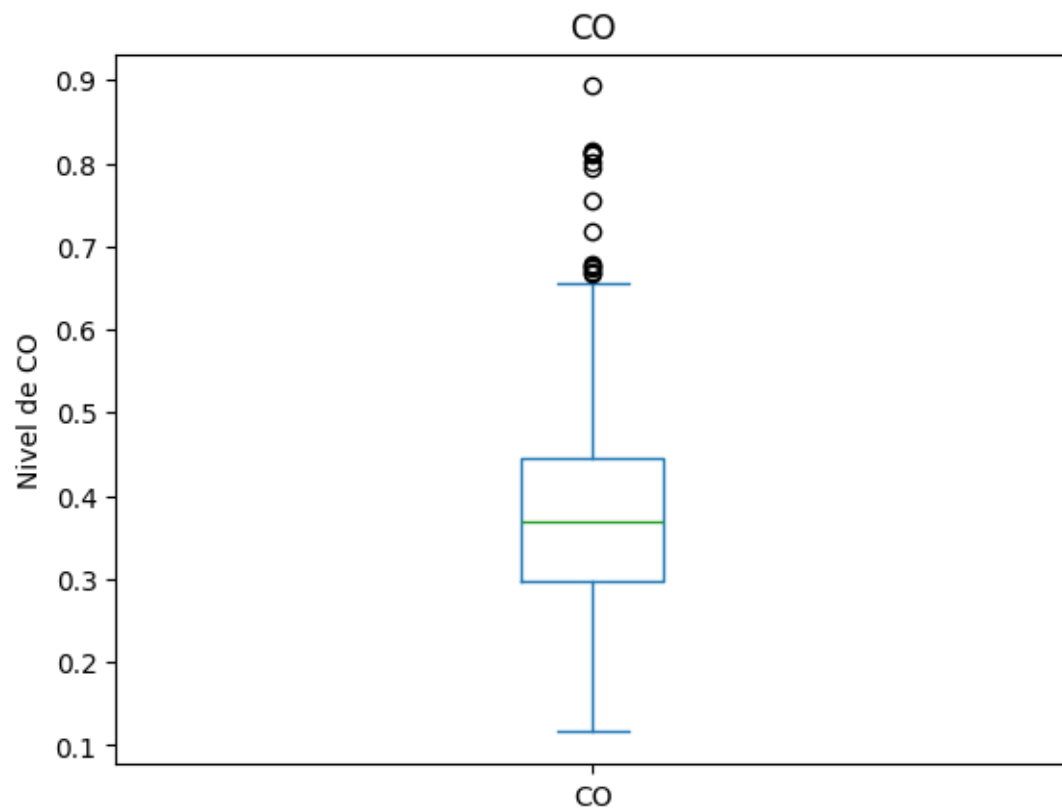
```

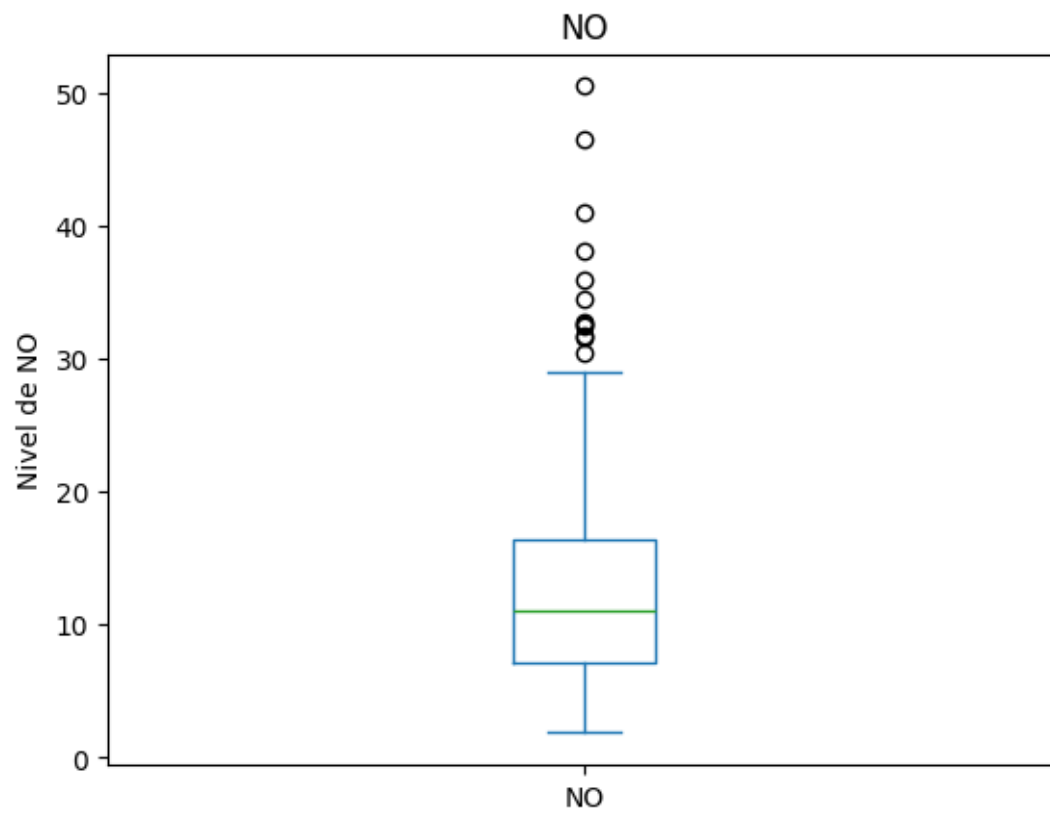
Name: SO2, dtype: float64

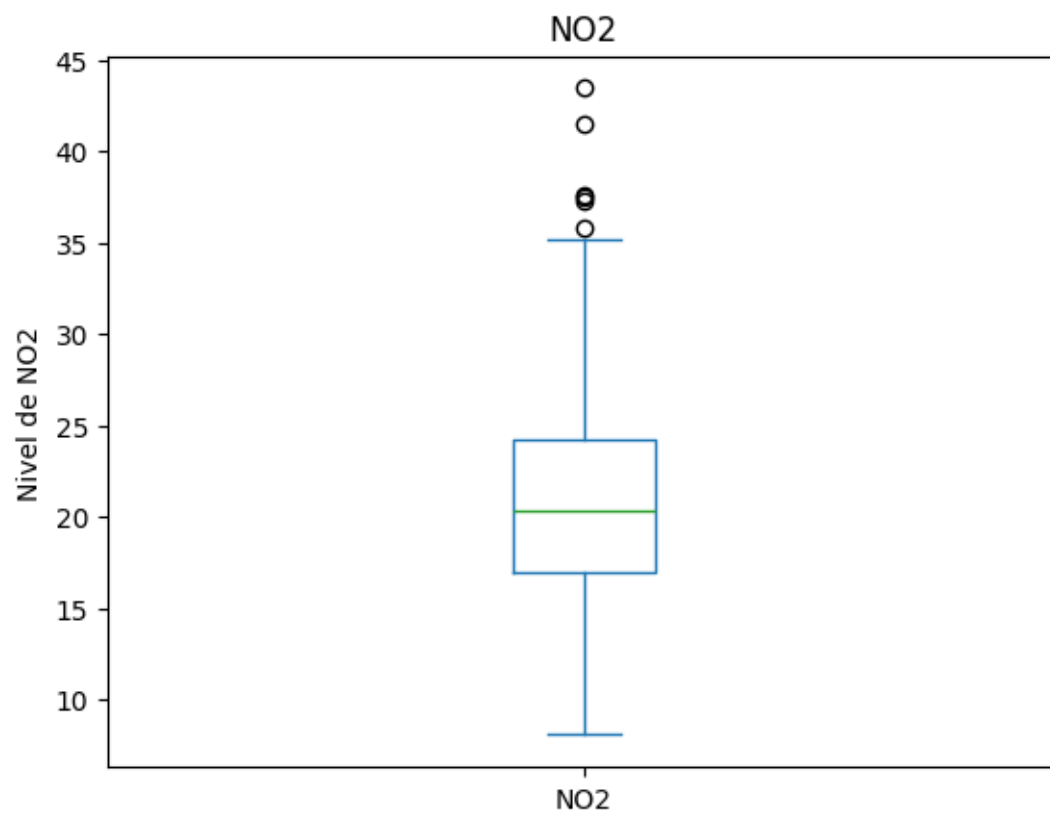
Asimetría: 4.461196942269091

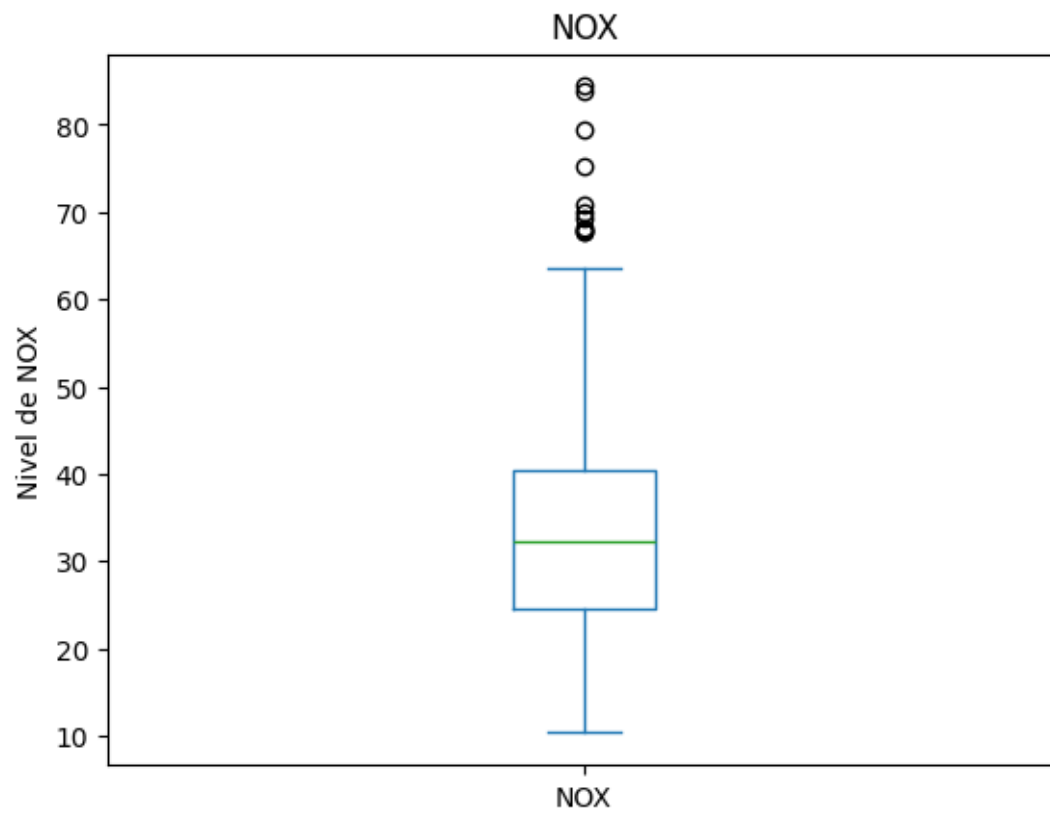
Curtosis: 31.87671519245468

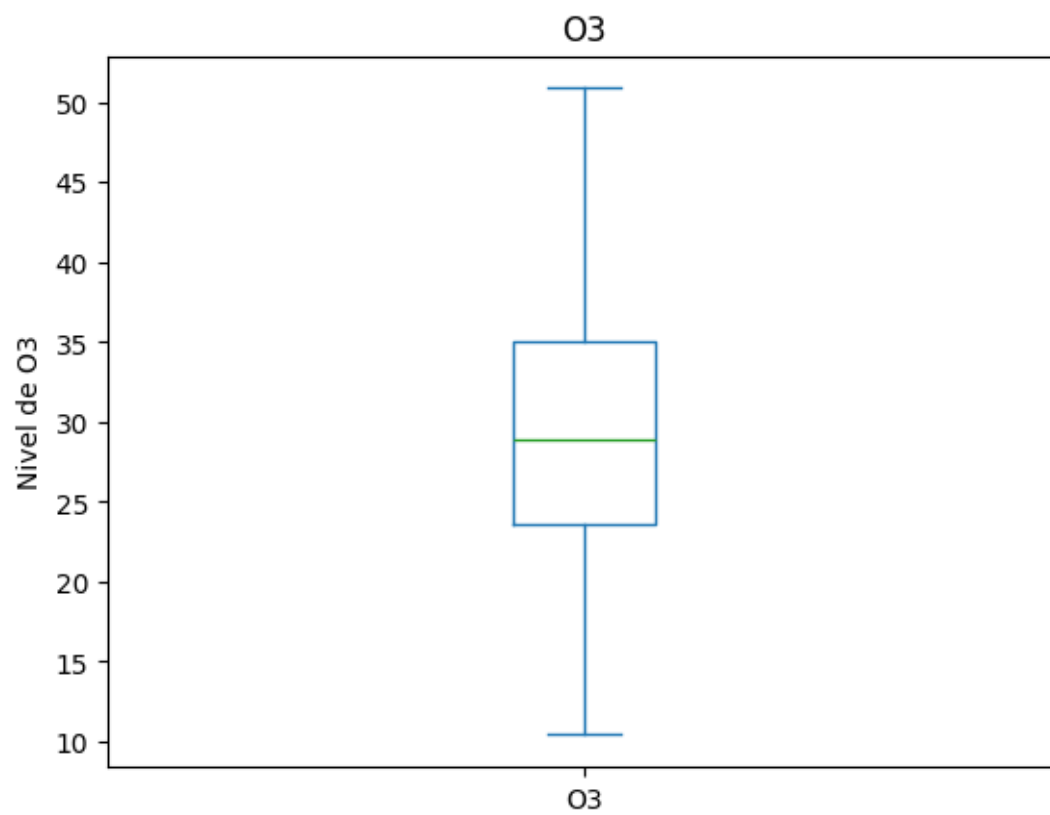
```
[112]: for column in rama2021.columns:  
        rama2021[column].plot.box()  
        pyplot.title(column)  
        pyplot.ylabel(f"Nivel de {column}")  
        pyplot.show()
```

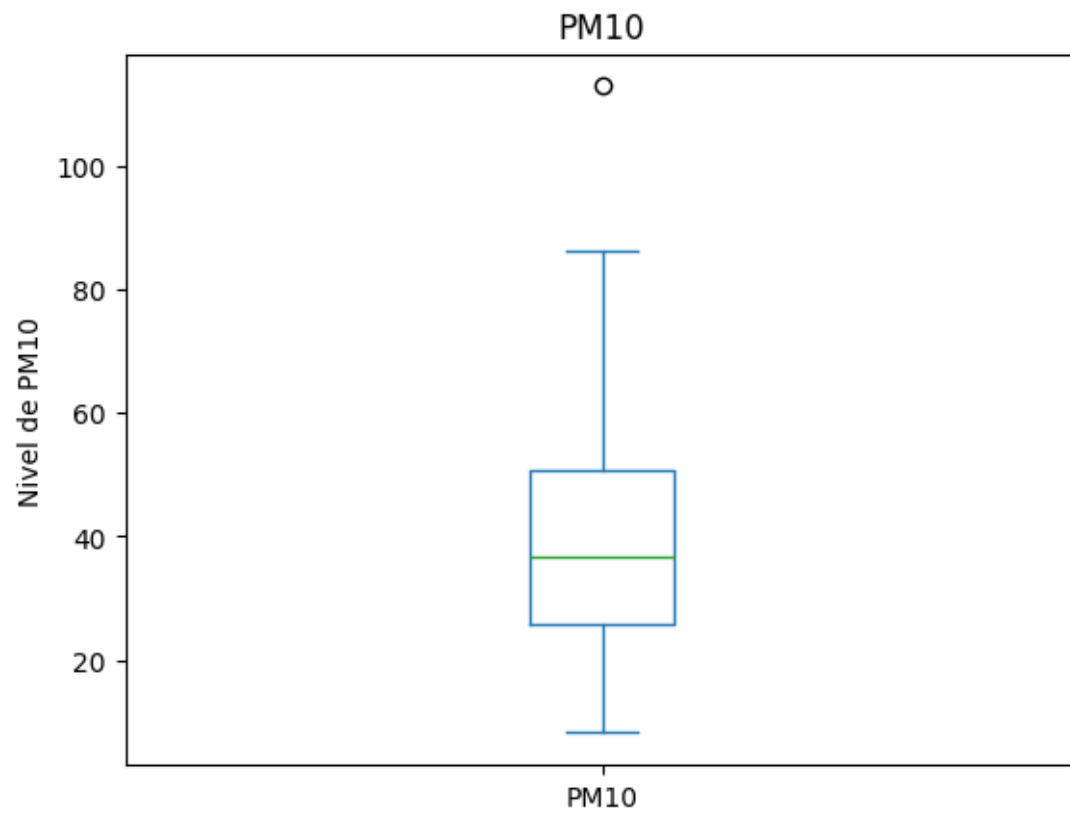


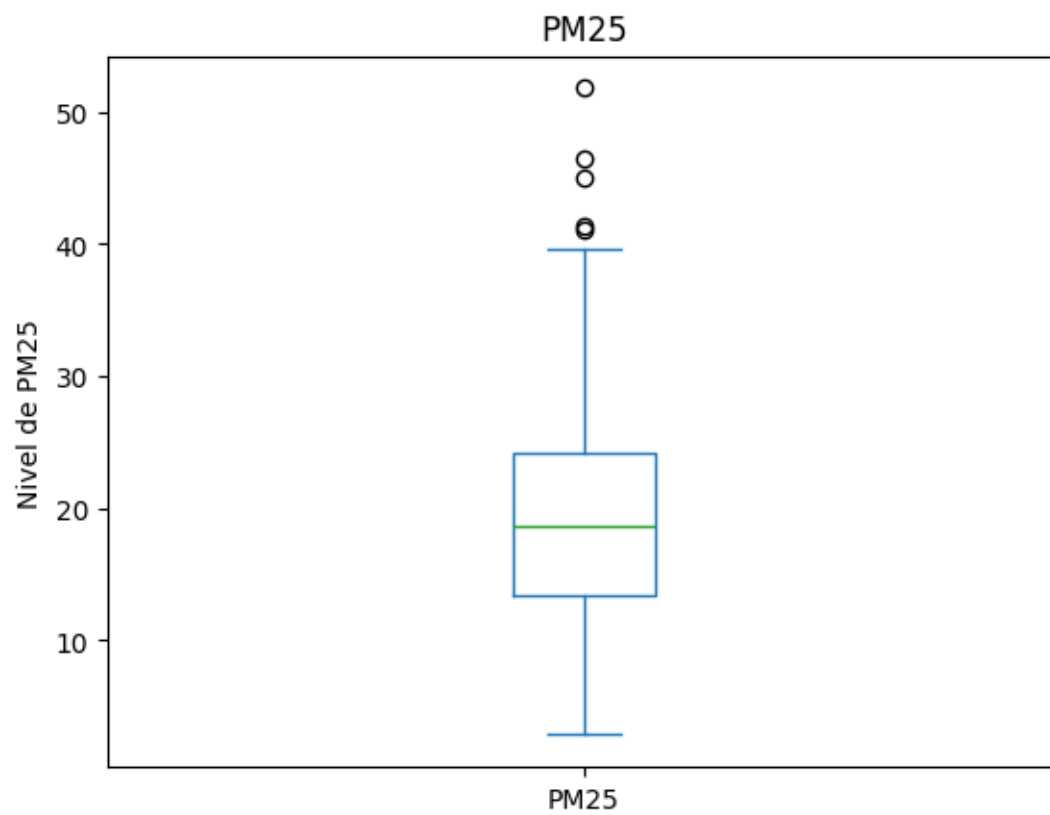


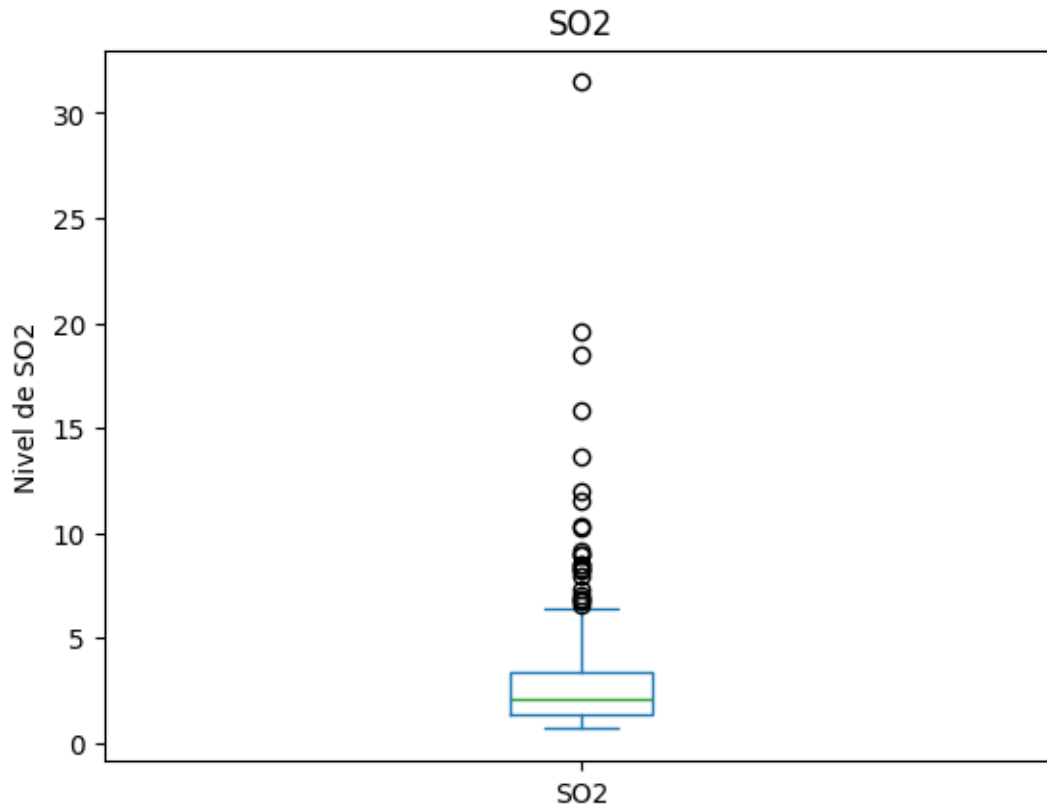












Para ver cuál fue el contaminante más detectado podemos normalizar los datos para ver cuál tiene un valor más alto, pero debemos quitar los puntos atípicos primero

```
[113]: rama2021_normalizado = rama2021.copy()

for column in rama2021.columns:
    x = rama2021_normalizado[column]

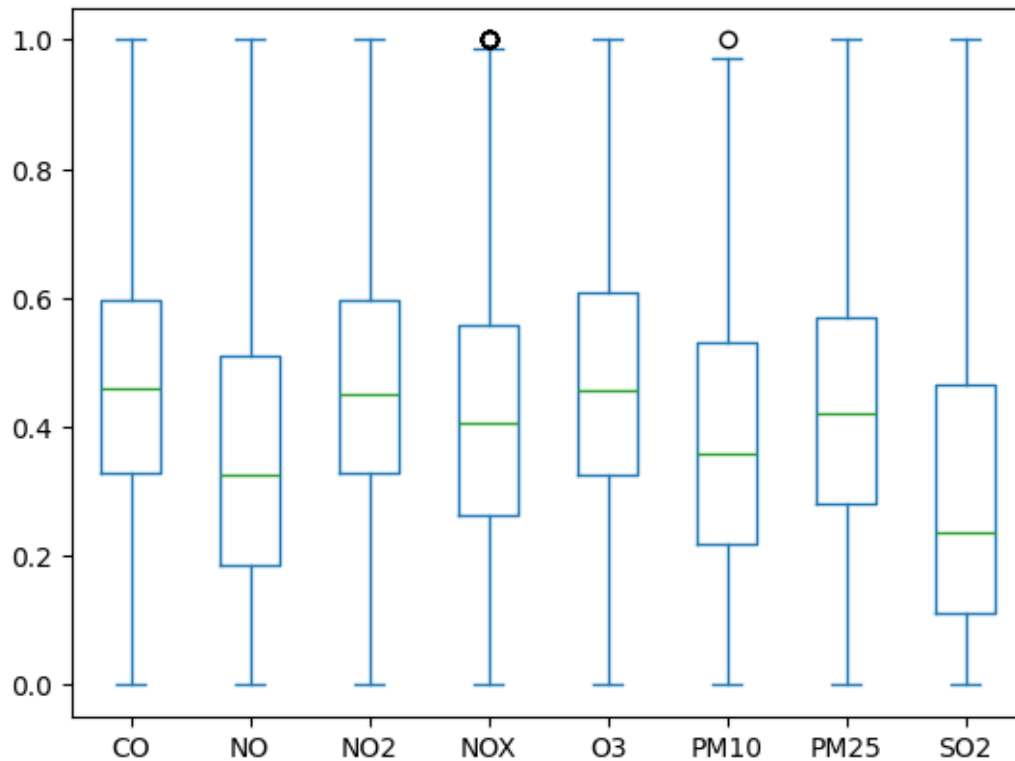
    Q1 = x.quantile(0.25)
    Q3 = x.quantile(0.75)
    IQR = Q3 - Q1
    xmin = Q1 - 1.5 * IQR
    xmax = Q3 + 1.5 * IQR
    x.loc[x < xmin] = xmin
    x.loc[x > xmax] = xmax

    x = (x - x.min()) / (x.max() - x.min())

    rama2021_normalizado[column] = x

rama2021_normalizado.plot.box()
```

```
pyplot.show()
```



Los compuestos más detectados son monóxido de carbono, dióxido de nitrógeno y ozono.

```
[116]: reporte = []

for column in rama2021.columns:
    x = rama2021[column]
    r_value, p_value = shapiro(x)
    reporte.append((column, r_value, p_value, p_value > 0.05))

pandas.DataFrame(reporte, columns=["Contaminante", "r-value", "p-value", "H0"])
```

```
[116]:
```

	Contaminante	r-value	p-value	H0
0	CO	0.949222	7.010633e-10	False
1	NO	0.907053	3.505852e-14	False
2	NO2	0.975581	7.778483e-06	False
3	NOX	0.944904	2.070274e-10	False
4	O3	0.986353	1.630253e-03	False
5	PM10	0.972192	1.840620e-06	False
6	PM25	0.975785	8.509463e-06	False
7	SO2	0.622303	1.550140e-27	False

Observamos que ningún contaminante acepta la hipótesis nula de normalidad, por lo que asumimos que son no-normales y debemos usar una correlación de Spearman en lugar de Pearson.

```
[117]: rama2021.corr(method="spearman")
```

```
[117]:
```

	CO	NO	NO2	NOX	O3	PM10	PM25	\
CO	1.000000	0.843404	0.873391	0.906800	-0.004116	0.504046	0.512779	
NO	0.843404	1.000000	0.785366	0.954812	-0.132850	0.484106	0.371816	
NO2	0.873391	0.785366	1.000000	0.924388	0.143831	0.645877	0.630509	
NOX	0.906800	0.954812	0.924388	1.000000	0.011048	0.607047	0.529378	
O3	-0.004116	-0.132850	0.143831	0.011048	1.000000	0.539238	0.640220	
PM10	0.504046	0.484106	0.645877	0.607047	0.539238	1.000000	0.895745	
PM25	0.512779	0.371816	0.630509	0.529378	0.640220	0.895745	1.000000	
S02	0.299368	0.295461	0.478446	0.404369	0.301324	0.611600	0.588893	


```

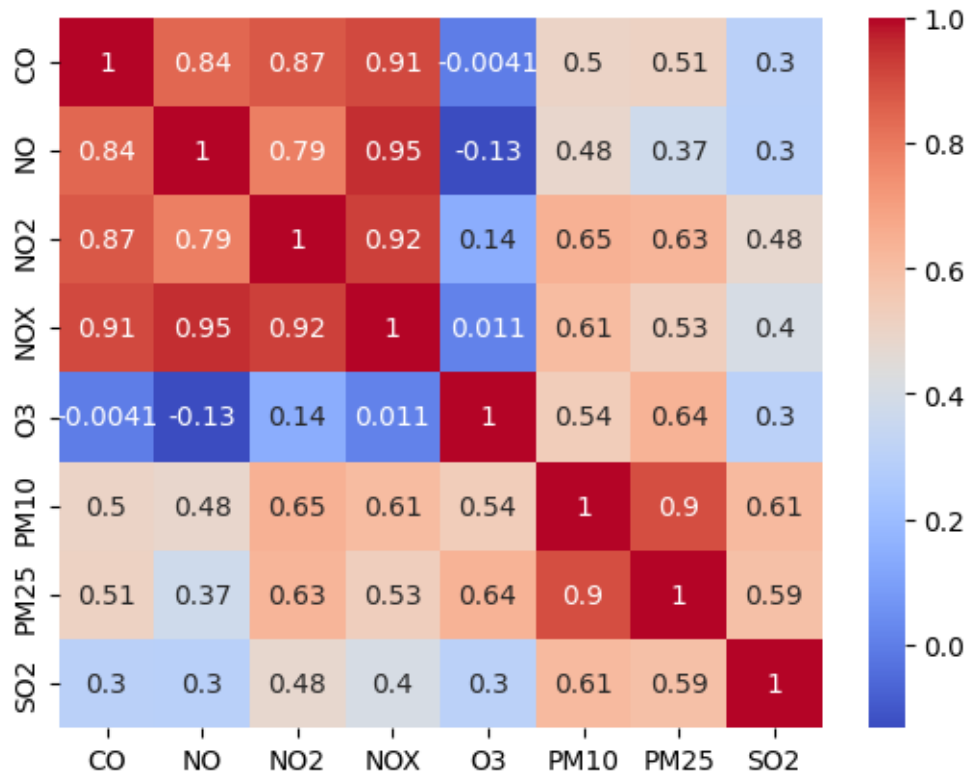
          S02
CO      0.299368
NO      0.295461
NO2     0.478446
NOX     0.404369
O3      0.301324
PM10    0.611600
PM25    0.588893
S02     1.000000

```

```
[120]: import seaborn

seaborn.heatmap(rama2021.corr(method="spearman"), cmap="coolwarm", annot=True)
```

```
[120]: <Axes: >
```



```
[124]: import pingouin
```

```
pingouin.pairwise_corr(rama2021).sort_values("r", ascending=False)
```

```
[124]:
```

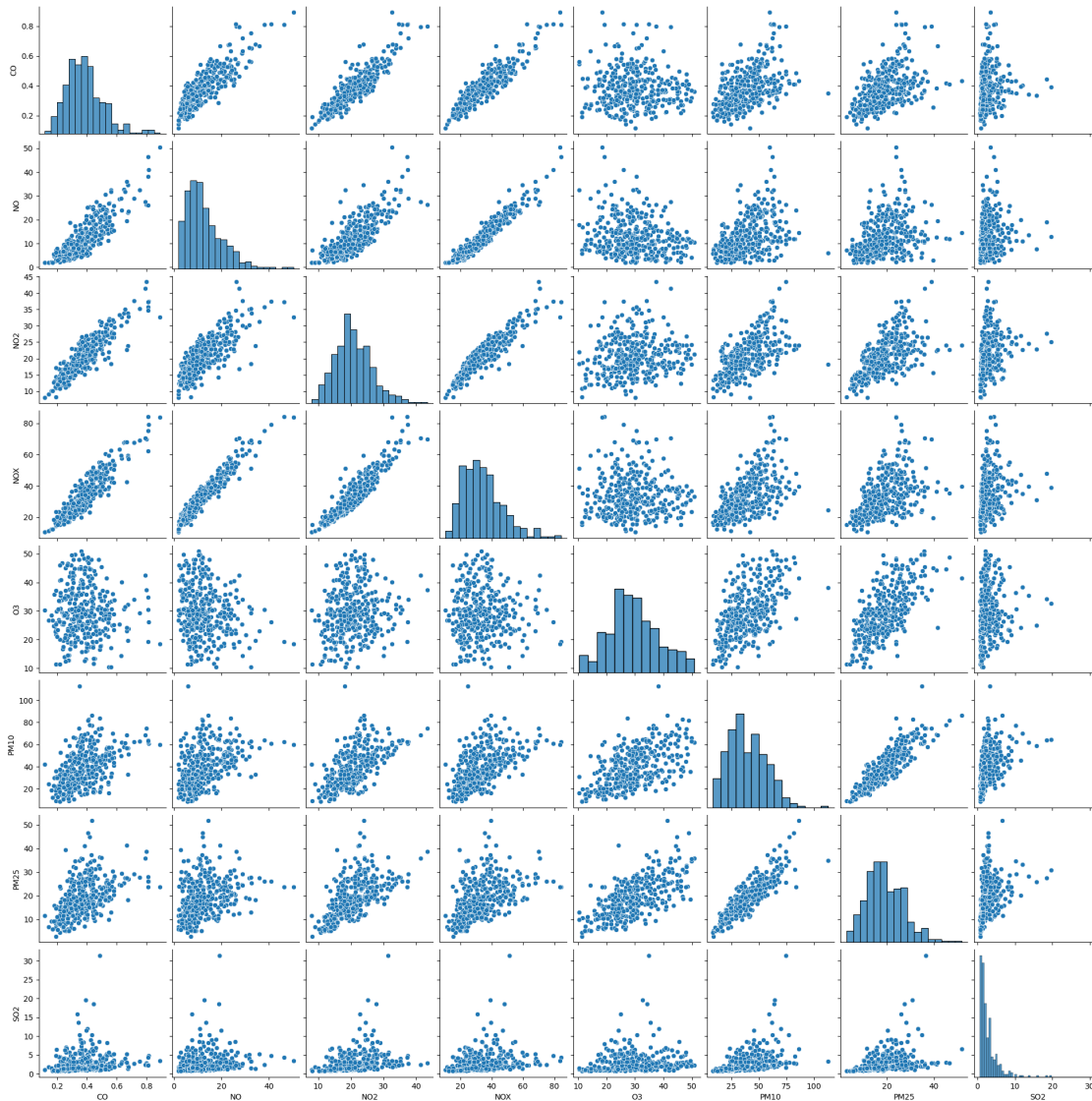
	X	Y	method	alternative	n	r	CI95%	\
8	NO	NOX	pearson	two-sided	365	0.957376	[0.95, 0.97]	
13	NO2	NOX	pearson	two-sided	365	0.927193	[0.91, 0.94]	
2	CO	NOX	pearson	two-sided	365	0.919823	[0.9, 0.93]	
1	CO	NO2	pearson	two-sided	365	0.889872	[0.87, 0.91]	
25	PM10	PM25	pearson	two-sided	365	0.883877	[0.86, 0.9]	
0	CO	NO	pearson	two-sided	365	0.857711	[0.83, 0.88]	
7	NO	NO2	pearson	two-sided	365	0.781486	[0.74, 0.82]	
23	O3	PM25	pearson	two-sided	365	0.653953	[0.59, 0.71]	
15	NO2	PM10	pearson	two-sided	365	0.616943	[0.55, 0.68]	
16	NO2	PM25	pearson	two-sided	365	0.585406	[0.51, 0.65]	
19	NOX	PM10	pearson	two-sided	365	0.559526	[0.48, 0.63]	
22	O3	PM10	pearson	two-sided	365	0.546269	[0.47, 0.61]	
4	CO	PM10	pearson	two-sided	365	0.486226	[0.4, 0.56]	
5	CO	PM25	pearson	two-sided	365	0.475767	[0.39, 0.55]	
20	NOX	PM25	pearson	two-sided	365	0.464237	[0.38, 0.54]	
10	NO	PM10	pearson	two-sided	365	0.435302	[0.35, 0.51]	
26	PM10	SO2	pearson	two-sided	365	0.426355	[0.34, 0.51]	

27	PM25	S02	pearson	two-sided	365	0.378781	[0.29, 0.46]
17	N02	S02	pearson	two-sided	365	0.315189	[0.22, 0.4]
11	N0	PM25	pearson	two-sided	365	0.305418	[0.21, 0.4]
21	NOX	S02	pearson	two-sided	365	0.246843	[0.15, 0.34]
12	N0	S02	pearson	two-sided	365	0.163643	[0.06, 0.26]
14	N02	03	pearson	two-sided	365	0.143000	[0.04, 0.24]
24	03	S02	pearson	two-sided	365	0.138879	[0.04, 0.24]
6	C0	S02	pearson	two-sided	365	0.134400	[0.03, 0.23]
3	C0	03	pearson	two-sided	365	-0.021812	[-0.12, 0.08]
18	NOX	03	pearson	two-sided	365	-0.022729	[-0.13, 0.08]
9	N0	03	pearson	two-sided	365	-0.173004	[-0.27, -0.07]

	p-unc	BF10	power
8	7.277622e-198	3.322e+193	1.000000
13	7.126383e-157	5.886e+152	1.000000
2	1.426633e-149	3.25e+145	1.000000
1	8.886360e-126	7.287e+121	1.000000
25	7.581649e-122	9.036e+117	1.000000
0	6.427693e-107	1.326e+103	1.000000
7	2.293188e-76	5.994e+72	1.000000
23	6.600362e-46	3.644e+42	1.000000
15	1.174811e-39	2.345e+36	1.000000
16	5.971849e-35	5.153e+31	1.000000
19	1.866443e-31	1.801e+28	1.000000
22	8.825105e-30	3.982e+26	1.000000
4	4.673732e-23	9.166e+19	1.000000
5	5.137210e-22	8.629e+18	1.000000
20	6.572201e-21	7.004e+17	1.000000
10	2.616885e-18	1.934e+15	1.000000
26	1.488477e-17	3.503e+14	1.000000
27	6.741578e-14	9.075e+10	1.000000
17	7.334132e-10	1.045e+07	0.999990
11	2.554397e-09	3.113e+06	0.999974
21	1.807656e-06	5559.882	0.997756
12	1.707753e-03	8.802	0.882132
14	6.205655e-03	2.731	0.783167
24	7.882320e-03	2.205	0.758922
6	1.015318e-02	1.761	0.731012
3	6.779011e-01	0.071	0.069960
18	6.651565e-01	0.072	0.071698
9	9.035250e-04	15.807	0.914540

```
[125]: seaborn.pairplot(rama2021)
```

```
[125]: <seaborn.axisgrid.PairGrid at 0x15ae3c8f0>
```



```
[127]: redmet2021 = pandas.read_csv("redmet_2021.csv", index_col=0, parse_dates=True)

redmet2021.head()
```

```
[127]:
```

	RH	TMP	WDR	WSP
fecha				
2021-01-01	32.645714	12.519203	162.673611	2.697743
2021-01-02	29.782353	12.918297	166.820467	1.954039
2021-01-03	32.780488	14.381852	179.656881	1.960917
2021-01-04	36.739563	13.840727	190.878403	1.705808
2021-01-05	39.845996	13.591215	163.173832	1.834019

```
[131]: datos2021 = rama2021.join(redmet2021)

datos2021.head()
```

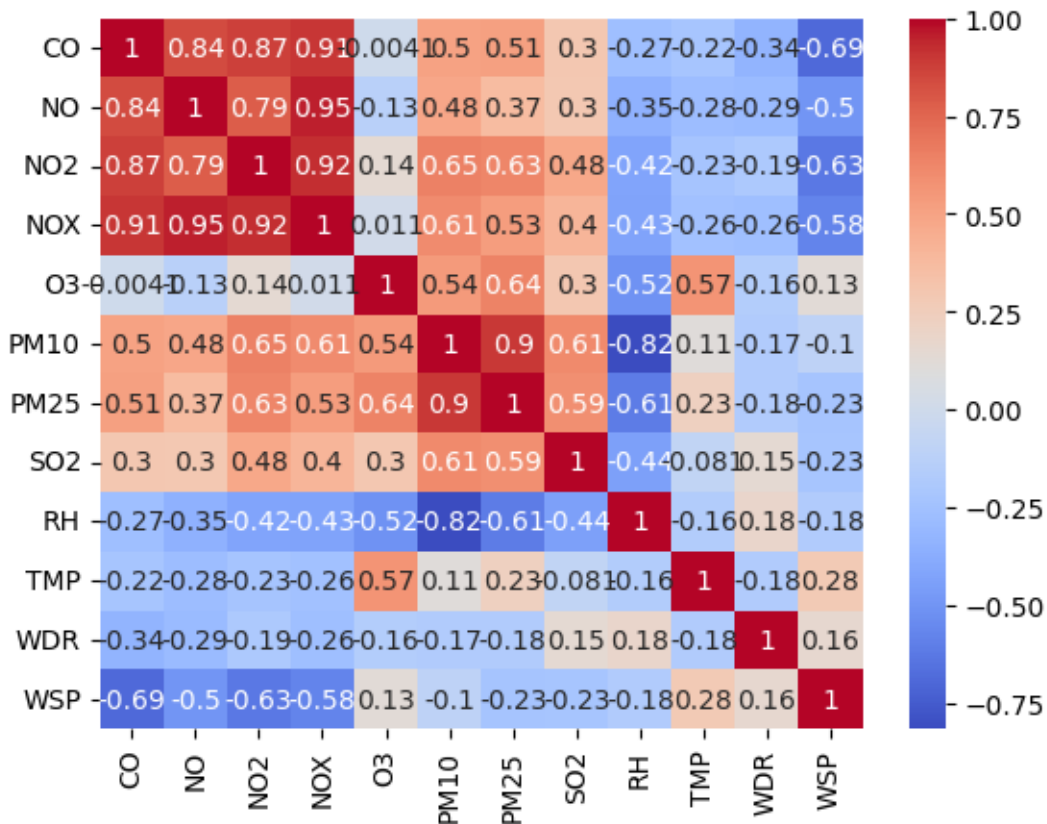
```
[131]:
```

	CO	NO	NO2	NOX	O3	PM10 \
fecha						
2021-01-01	0.417782	5.712963	16.977273	23.261574	29.765494	54.327273
2021-01-02	0.515087	14.289894	26.782609	42.090426	33.647969	48.196481
2021-01-03	0.446905	12.328571	22.773256	35.714286	28.135823	39.091988
2021-01-04	0.580398	22.323040	31.306483	54.418052	28.824532	52.486239
2021-01-05	0.533105	19.386091	29.376218	49.995204	28.941374	55.916667

	PM25	SO2	RH	TMP	WDR	WSP
fecha						
2021-01-01	28.594855	1.125912	32.645714	12.519203	162.673611	2.697743
2021-01-02	23.465409	1.725000	29.782353	12.918297	166.820467	1.954039
2021-01-03	16.423676	1.548983	32.780488	14.381852	179.656881	1.960917
2021-01-04	23.160000	3.410019	36.739563	13.840727	190.878403	1.705808
2021-01-05	22.694444	5.251366	39.845996	13.591215	163.173832	1.834019

```
[132]: seaborn.heatmap(datos2021.corr(method="spearman"), cmap="coolwarm", annot=True)
```

```
[132]: <Axes: >
```



Observamos que hay alta correlación negativa entre algunos compuestos contaminantes y factores meteorológicos como la humedad relativa (RH) y el contaminante PM_{10} que tiene una correlación de -0.82.

Conclusiones

Al analizar los contaminantes hemos observado que estos no se comportan de forma normal, por lo que hemos analizado la correlación entre estos mediante el método de Spearman.

Además al cruzar la información con los factores meteorológicos, vemos que cuando el valor de los factores meteorológicos aumenta como la humedad relativa (RH), la temperatura (TMP), la dirección del viento (WDR) y la velocidad del viento (WSP) aumentan, algunos contaminantes disminuyen (en correlación negativa).

Esto nos muestra que la contaminación aumenta cuando los factores meteorológicos son normales y nos permite precisar hipótesis como que aumentar la humedad relativa podría ayudar a disminuir la contaminación. Sin embargo, esto no es concluyente ni debe ser causal, por ejemplo, podría ser que las lecturas se alteren con más humedad.