

```
import math
import statistics
import scipy.stats
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import re
from sklearn import preprocessing
import seaborn as sns
from unicode import unicode
```

```
df_sai=pd.read_excel('datos_finales.xlsx')
df_sai
```

	Equipo int64 100171 - 10031587	calificacion_del_... 1 - 5	capacitancia_total f -0.110129117 - 0.22...	factor_de_poten... -0.999572824 - 3....	factor_de_poten... -0.989348659 - 3....	capacitancia_tot... 469.443 - 10036.0	factor_de_poten... 0.08 - 0.67	factor_de_poten... 0.03 - 0.64
0	100171	1	-0.000415105	0.847826087	0.816216216	794.65	0.34	0.336
1	100172	1	0.002779849	1.205479452	1.393103448	793.61	0.322	0.347
2	100173	2	-0.004502019	1.129032258	1.137254902	793.83	0.33	0.327
3	100189	1	0.062631254	0.368421053	0.368421053	860.2	0.26	0.26
4	100190	1	0.064350064	0.736842105	0.736842105	827.0	0.33	0.33
5	100191	1	0.058713304	0.75	0.75	847.5	0.35	0.35
6	100195	1	0.058794588	0.04	0.066666667	860.8	0.26	0.256
7	100196	1	0.059877301	0.272	0.252	863.8	0.318	0.313
8	100197	1	0.064709518	0.336363636	0.260869565	861.35	0.294	0.29
9	100207	1	0.06031407	0.26	0.294736842	844.01	0.252	0.246

```
df_sai_copy1=df_sai.copy()
```

```
df_sai_copy1["CHANGED_estado_cajetin"] = df_sai_copy1["estado_cajetin"].map({'Malo':1,
                                                                                'Regular':3,
                                                                                'Bueno':5})

df_sai_copy1["CHANGED_estado_porcelana"] = df_sai_copy1["estado_de_la_porcelana"].map({'Roto':1,
                                                                                       'Malo':1,
                                                                                       'Despicado':3,
                                                                                       'Despicada':3,
                                                                                       'Bueno':5})

df_sai_copy1["CHANGED_fuga_de_aceite"] = df_sai_copy1["fuga_de_aceite"].map({'No':5,
                                                                                'Mancha':3,
                                                                                'Goteo intermitente':1,
                                                                                'No hay':5,
                                                                                'goteo intermitente':1,
                                                                                'goteo continuo':0})

df_sai_copy1["CHANGED_inspeccion_diafragma"] = df_sai_copy1["inspeccion_diafragma"].map({'Bueno':5,
                                                                                       'Regular':3,
                                                                                       'Malo':1})

df_sai_copy1["CHANGED_inspeccion_visual_general"] = df_sai_copy1["inspeccion_visual_general"].map({'Bueno':5,
                                                                                       'Regular':3,
                                                                                       'Malo':1})

df_sai_copy1["CHANGED_nivel_de_aceite"] = df_sai_copy1["nivel_de_aceite"].map({'Bajo':1,
                                                                                'Normal':5,
```

```
'Medio':3,
'Bueno':5,
'medio':3})
```

df_sai_copy1

	Equipo int64 100171 - 10031587	calificacion_del_... 1 - 5	capacitancia_total i -0.110129117 - 0.22...	factor_de_poten... -0.999572824 - 3....	factor_de_poten... -0.989348659 - 3....	capacitancia_tot... 469.443 - 10036.0	factor_de_poten... 0.08 - 0.67	factor_de_poten... 0.03 - 0.64
0	100171	1	-0.000415105	0.847826087	0.816216216	794.65	0.34	0.336
1	100172	1	0.002779849	1.205479452	1.393103448	793.61	0.322	0.347
2	100173	2	-0.004502019	1.129032258	1.137254902	793.83	0.33	0.327
3	100189	1	0.062631254	0.368421053	0.368421053	860.2	0.26	0.26
4	100190	1	0.064350064	0.736842105	0.736842105	827.0	0.33	0.33
5	100191	1	0.058713304	0.75	0.75	847.5	0.35	0.35
6	100195	1	0.058794588	0.04	0.066666667	860.8	0.26	0.256
7	100196	1	0.059877301	0.272	0.252	863.8	0.318	0.313
8	100197	1	0.064709518	0.336363636	0.260869565	861.35	0.294	0.29
9	100207	1	0.06031407	0.26	0.294736842	844.01	0.252	0.246

```
#Se usara las columnas depuradas
#Se eliminaron los codigos de los equipos, nombres de modelos y las variables que fueron codificadas.
df_sai_copy2=df_sai_copy1.copy().drop(["Equipo",
                                         "estado_cajetin",
                                         "estado_de_la_porcelana",
                                         "fuga_de_aceite",
                                         "inspeccion_diafragma",
                                         "inspeccion_visual_general",
                                         "nivel_de_aceite"],axis=1)
```

df_sai_copy2

	calificacion_del_... 1 - 5	capacitancia_total i -0.110129117 - 0.22...	factor_de_poten... -0.999572824 - 3....	factor_de_poten... -0.989348659 - 3....	capacitancia_tot... 469.443 - 10036.0	factor_de_poten... 0.08 - 0.67	factor_de_poten... 0.03 - 0.64	medida_resisten... 25 - 651000
0	1	-0.000415105	0.847826087	0.816216216	794.65	0.34	0.336	50
1	1	0.002779849	1.205479452	1.393103448	793.61	0.322	0.347	81
2	2	-0.004502019	1.129032258	1.137254902	793.83	0.33	0.327	158
3	1	0.062631254	0.368421053	0.368421053	860.2	0.26	0.26	705
4	1	0.064350064	0.736842105	0.736842105	827.0	0.33	0.33	650
5	1	0.058713304	0.75	0.75	847.5	0.35	0.35	4190
6	1	0.058794588	0.04	0.066666667	860.8	0.26	0.256	93
7	1	0.059877301	0.272	0.252	863.8	0.318	0.313	1440
8	1	0.064709518	0.336363636	0.260869565	861.35	0.294	0.29	25
9	1	0.06031407	0.26	0.294736842	844.01	0.252	0.246	3570

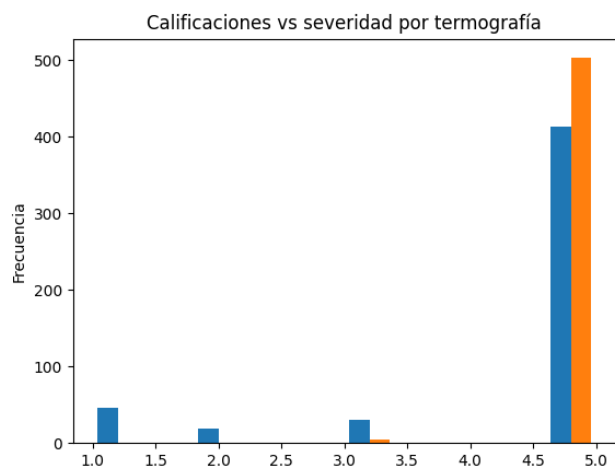
```
plt.title('Histograma de calificaciones')

plt.hist(df_sai_copy2['calificacion_del_equipo'], color='lightgreen')
```

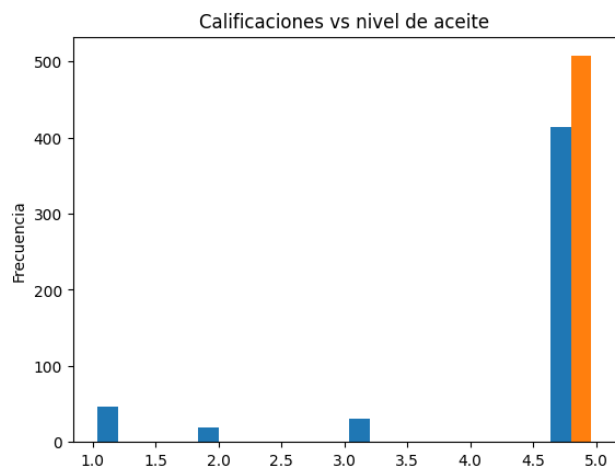
```
plt.ylabel('Frecuencia')  
plt.savefig("hist_calificaciones.jpg")
```



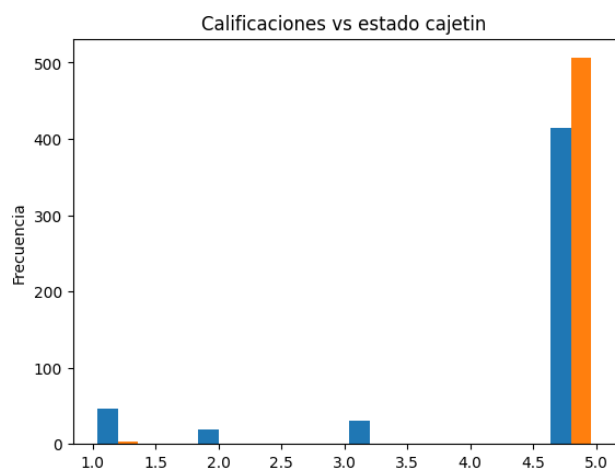
```
plt.title('Calificaciones vs severidad por termografía')  
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['severidad_por_termografia']], label=['Variable 1', 'Variable 2'])  
plt.ylabel('Frecuencia')  
plt.savefig("calificaciones vs severidad.jpg")
```



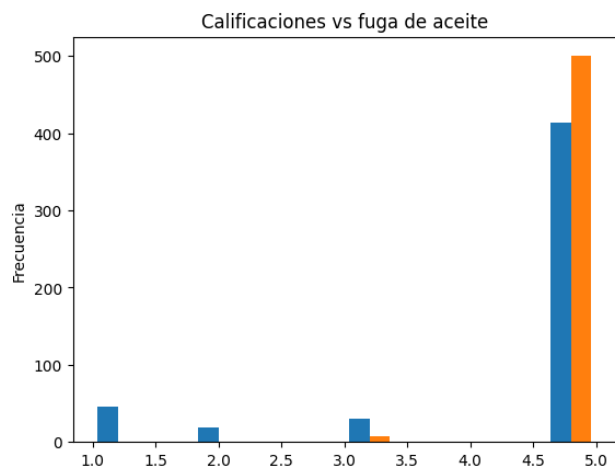
```
plt.title('Calificaciones vs nivel de aceite')  
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_nivel_de_aceite']], label=['Variable 1', 'Variable 2'])  
plt.ylabel('Frecuencia')  
plt.savefig("calificaciones vs nivel de aceite.jpg")
```



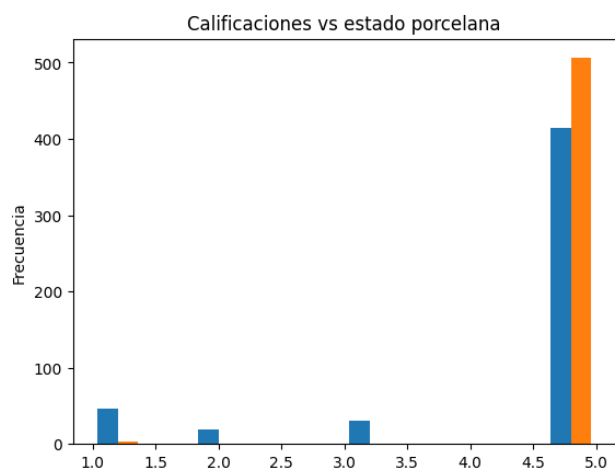
```
plt.title('Calificaciones vs estado cajetin')
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_estado_cajetin']], label=['Variable 1', 'Variable 2'])
plt.ylabel('Frecuencia')
plt.savefig("calificaciones vs estado_cajetin.jpg")
```



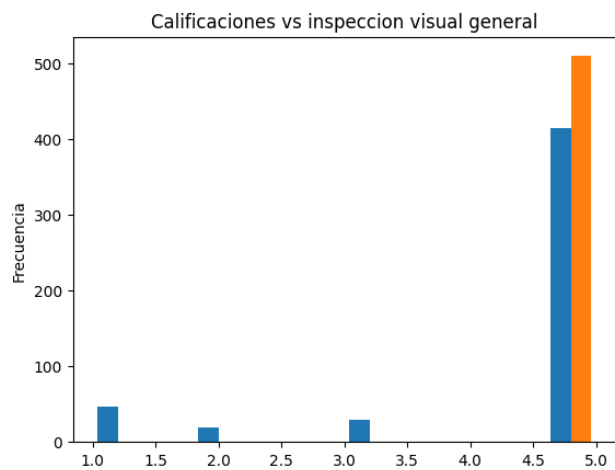
```
plt.title('Calificaciones vs fuga de aceite')
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_fuga_de_aceite']], label=['Variable 1', 'Variable 2'])
plt.ylabel('Frecuencia')
plt.savefig("calificaciones vs fuga_de_aceite.jpg")
```



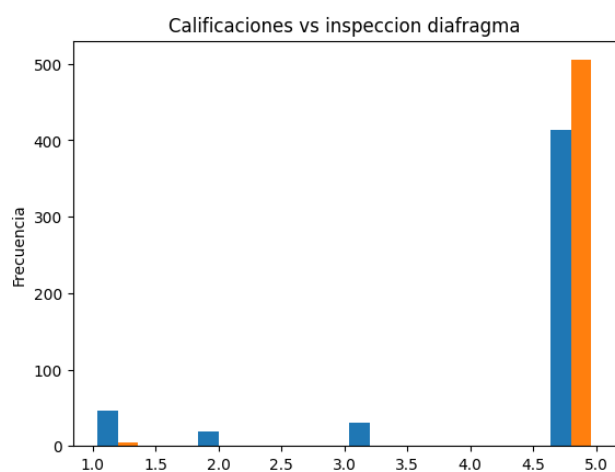
```
plt.title('Calificaciones vs estado porcelana')
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_estado_porcelana']], label=['Variable 1', 'Variable 2'])
plt.ylabel('Frecuencia')
plt.savefig("calificaciones vs estado_porcelana.jpg")
```



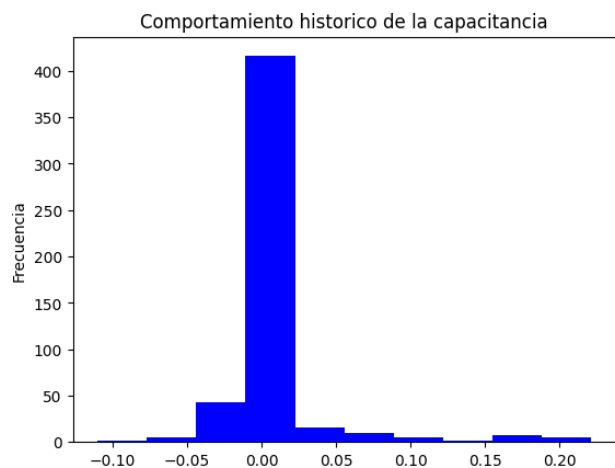
```
plt.title('Calificaciones vs inspeccion visual general')
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_inspeccion_visual_general']], label=['Variable 1', 'Variable 2'])
plt.ylabel('Frecuencia')
plt.savefig("calificaciones vs inspeccion_visual_general.jpg")
```



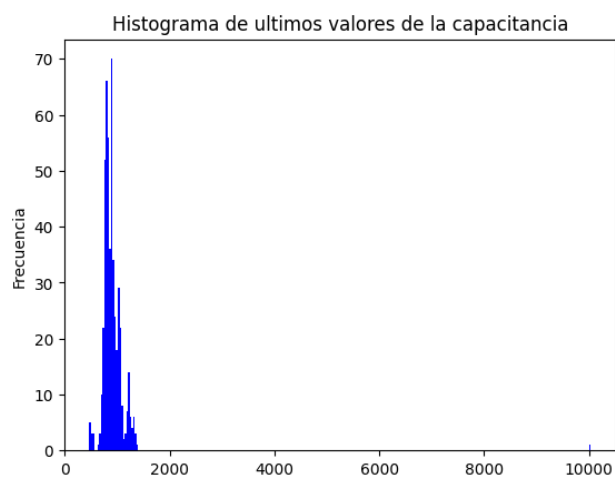
```
plt.title('Calificaciones vs inspeccion diafragma')
plt.hist([df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['CHANGED_inspeccion_diafragma']], label=['Variable 1', 'Variable 2'])
plt.ylabel('Frecuencia')
plt.savefig("calificaciones vs inspeccion_diafragma.jpg")
```



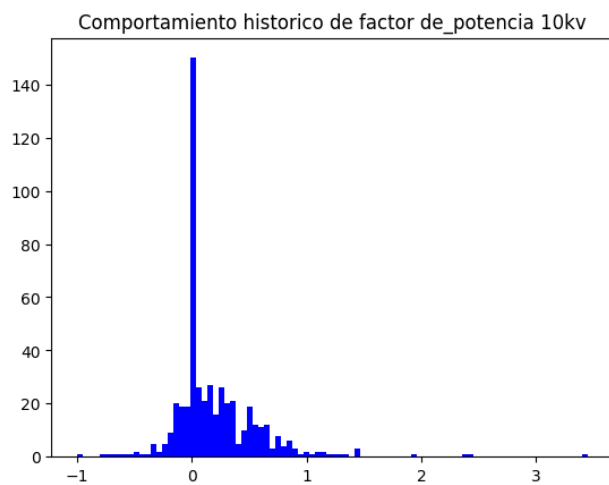
```
plt.title('Comportamiento historico de la capacitancia')
plt.hist(df_sai_copy2['capacitancia_total'], bins=10, color='Blue')
plt.ylabel('Frecuencia')
plt.savefig("Comportamiento_historico_de_la_capacitancia.jpg")
```



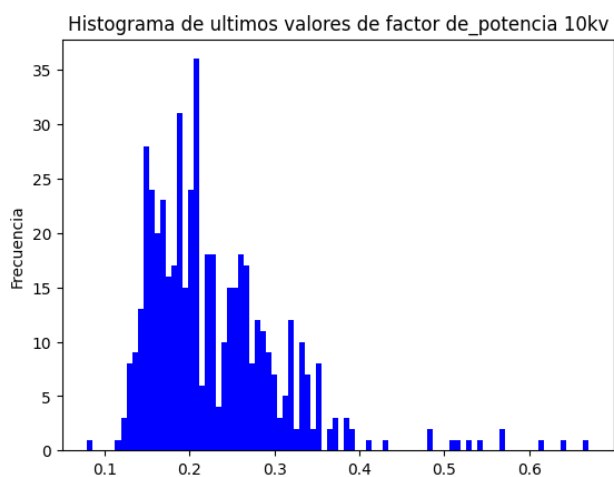
```
plt.title('Histograma de ultimos valores de la capacitancia')
plt.hist(df_sai_copy2['capacitancia_total_ultimo'], bins=300, color='Blue')
plt.ylabel('Frecuencia')
plt.savefig("ultmos_valores_la_capacitancia.jpg")
```



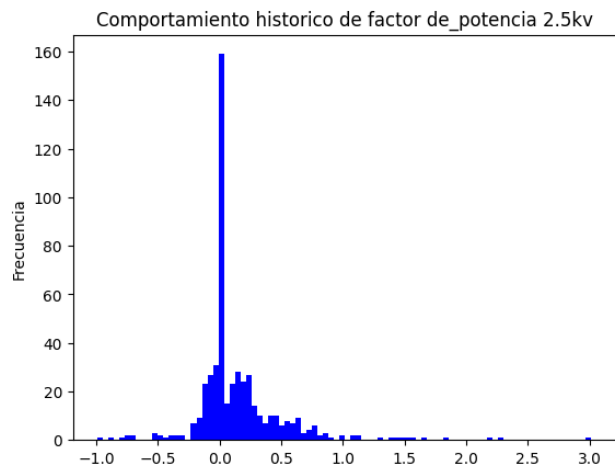
```
plt.title('Comportamiento historico de factor de potencia 10kv')
plt.hist(df_sai_copy2['factor_de_potencia_10kv_ctotal'], bins=90, color='Blue')
plt.savefig("Comportamiento historico de factor de potencia 10kv.jpg")
```



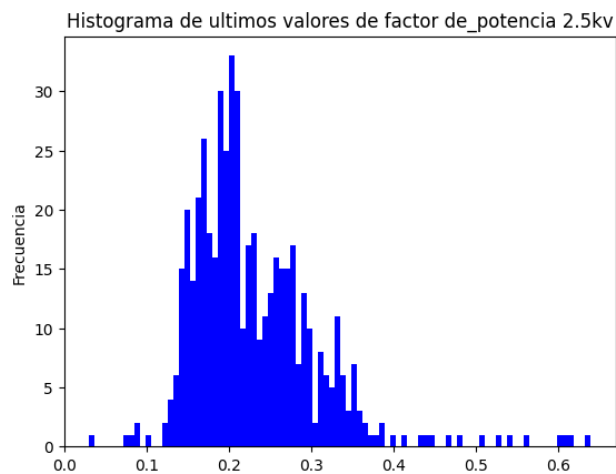
```
plt.title('Histograma de ultimos valores de factor de_potencia 10kv')
plt.hist(df_sai_copy2['factor_de_potencia_10kv_ctotal_ultimo'], bins=90, color='Blue')
plt.ylabel('Frecuencia')
plt.savefig("ultimos_valores_factor_de_potencia_10kv.jpg")
```



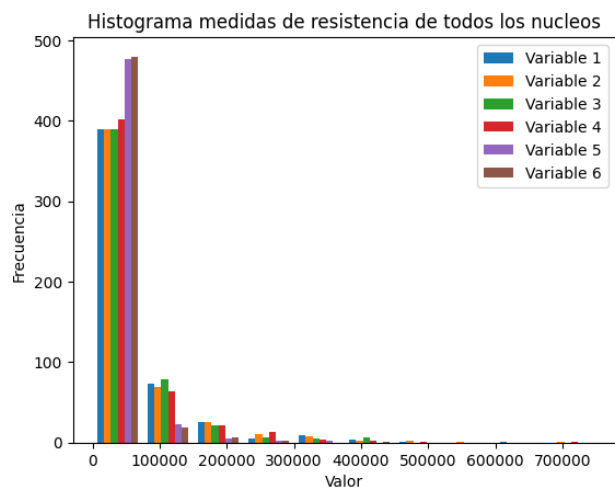
```
plt.title('Comportamiento historico de factor de_potencia 2.5kv')
plt.hist(df_sai_copy2['factor_de_potencia_2.5kv_ctotal'], bins=90, color='Blue')
plt.ylabel('Frecuencia')
plt.savefig("Comportamiento historico de factor de_potencia 2.5kv.jpg")
```

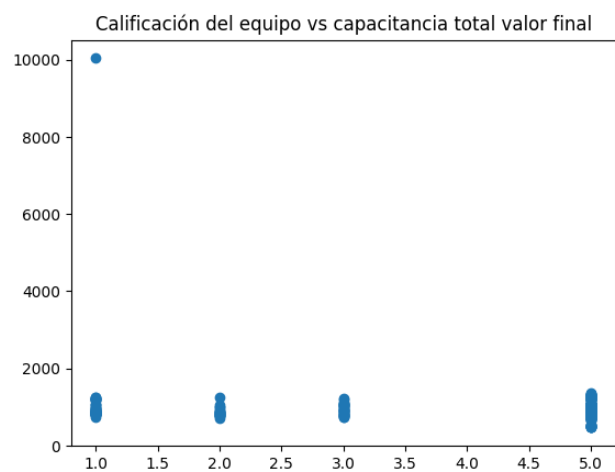
```
plt.title('Histograma de ultimos valores de factor de potencia 2.5kv')
plt.hist(df_sai_copy2['factor_de_potencia_2.5kv_ctotal_ultimo'], bins=90, color='Blue')
plt.ylabel('Frecuencia')
plt.savefig("ultimos_valores_factor_de_potencia_2.5kv.jpg")
```



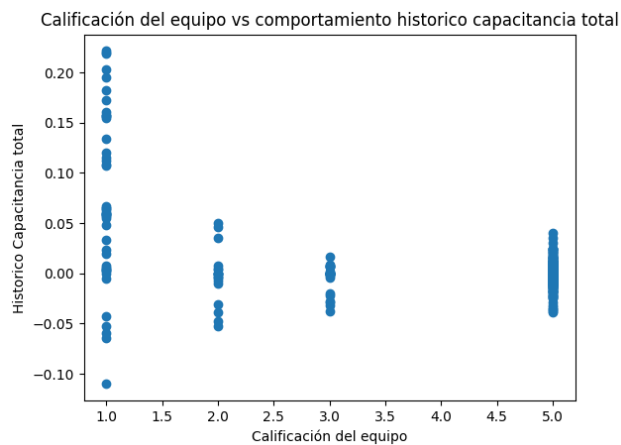
```
plt.hist([df_sai_copy2['medida_resistencia_aislamiento_nucleo_1_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_2_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_3_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_4_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_5_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_6_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_7_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_8_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_9_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_10_ultimo']], bins=10, color='red', edgecolor='black', rwidth=0.8)
plt.legend(loc='upper right')
# Agregar títulos y etiquetas de los ejes
plt.title('Histograma medidas de resistencia de todos los nucleos')
plt.xlabel('Valor')
plt.ylabel('Frecuencia')
plt.savefig("Histograma medidas de resistencia de todos los nucleos.jpg")
```



```
plt.title('Calificación del equipo vs capacitancia total valor final')
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['capacitancia_total_ultimo'])
plt.savefig("Calificación del equipo vs capacitancia total valor final.jpg")
```



```
plt.title('Calificación del equipo vs comportamiento historico capacitancia total')
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['capacitancia_total'])
plt.xlabel('Calificación del equipo')
plt.ylabel('Historico Capacitancia total')
plt.savefig("Calificación del equipo vs comportamiento historico capacitancia total.jpg")
```



```
x=df_sai_copy2['capacitancia_total']
y=df_sai_copy2['capacitancia_total_ultimo']

# Calcular los coeficientes de la línea de regresión
coefficients = np.polyfit(x, y, 1)

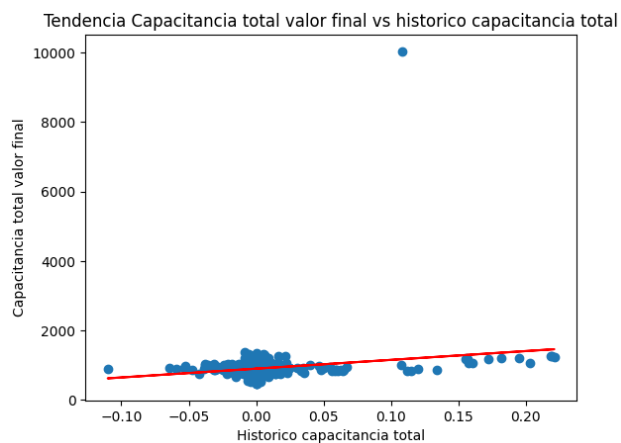
# Crear el scatter plot
plt.scatter(x, y)

# Calcular los valores de y correspondientes a los valores de x
y_fit = np.polyval(coefficients, x)

# Graficar la línea de tendencia
plt.plot(x, y_fit, 'r-')

# Agregar títulos y etiquetas de los ejes
plt.title('Tendencia Capacitancia total valor final vs historico capacitancia total')
plt.xlabel('Historico capacitancia total')
plt.ylabel('Capacitancia total valor final')

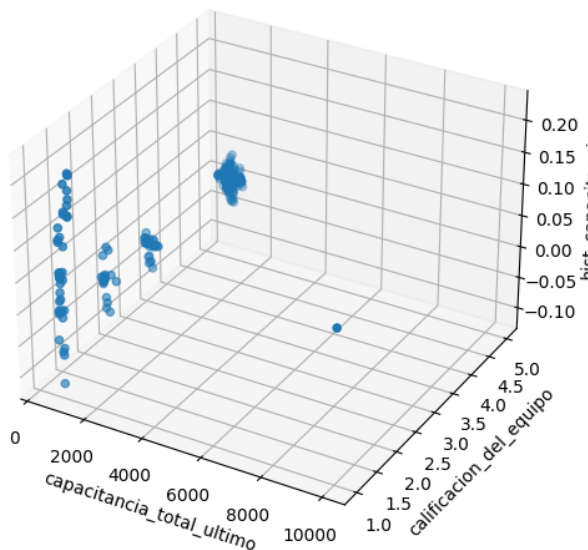
# Mostrar el gráfico
plt.show()
plt.savefig("tendencia Capacitancia total valor final vs comportamiento historico capacitancia total.jpg")
```



<Figure size 640x480 with 0 Axes>

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

# Crear el scatter plot 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
fig.set_size_inches(6, 6)
ax.set_xlabel('capacitancia_total_ultimo')
ax.set_ylabel('calificacion_del_equipo')
ax.set_zlabel('hist_capacitancia')
ax.scatter(df_sai_copy2['capacitancia_total_ultimo'], df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['hist_capacitancia'])
plt.savefig("Capacitancia total valor final vs comportamiento historico capacitancia total vs calificacion.jpg")
```



```
#plt.scatter(df_sai_copy2['factor_de_potencia_2.5kv_ctotal'], df_sai_copy2['factor_de_potencia_2.5kv_ctotal_ultimo'])
x=df_sai_copy2['factor_de_potencia_2.5kv_ctotal']
y=df_sai_copy2['factor_de_potencia_2.5kv_ctotal_ultimo']

# Calcular los coeficientes de la línea de regresión
coefficients = np.polyfit(x, y, 1)

# Crear el scatter plot
plt.scatter(x, y)

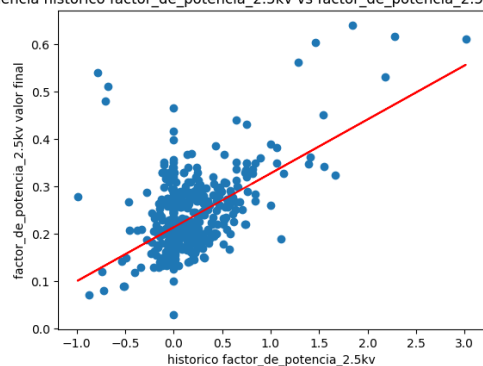
# Calcular los valores de y correspondientes a los valores de x
y_fit = np.polyval(coefficients, x)

# Graficar la línea de tendencia
plt.plot(x, y_fit, 'r-')

# Agregar títulos y etiquetas de los ejes
plt.title('Tendencia historico factor_de_potencia_2.5kv vs factor_de_potencia_2.5kv valor final')
plt.xlabel('historico factor_de_potencia_2.5kv')
plt.ylabel('factor_de_potencia_2.5kv valor final')

# Mostrar el gráfico
plt.show()
plt.savefig("tendencia factor_de_potencia_2.5kv valor final vs comportamiento historico factor_de_potencia_2.5kv.jpg")
```

Tendencia historico factor_de_potencia_2.5kv vs factor_de_potencia_2.5kv valor final



<Figure size 640x480 with 0 Axes>

```
#plt.scatter(df_sai_copy2['factor_de_potencia_10kv_ctotal'],df_sai_copy2['factor_de_potencia_10kv_ctotal_ultimo'])
#plt.scatter(df_sai_copy2['factor_de_potencia_2.5kv_ctotal'],df_sai_copy2['factor_de_potencia_2.5kv_ctotal_ultimo'])
x=df_sai_copy2['factor_de_potencia_10kv_ctotal']
y=df_sai_copy2['factor_de_potencia_10kv_ctotal_ultimo']

# Calcular los coeficientes de la línea de regresión
coefficients = np.polyfit(x, y, 1)

# Crear el scatter plot
plt.scatter(x, y)

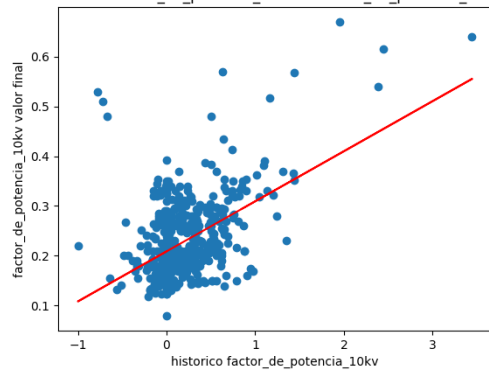
# Calcular los valores de y correspondientes a los valores de x
y_fit = np.polyval(coefficients, x)

# Graficar la línea de tendencia
plt.plot(x, y_fit, 'r-')

# Agregar títulos y etiquetas de los ejes
plt.title('Tendencia historico factor_de_potencia_10kv vs factor_de_potencia_10kv valor final')
plt.xlabel('historico factor_de_potencia_10kv')
plt.ylabel('factor_de_potencia_10kv valor final')

# Mostrar el gráfico
plt.show()
plt.savefig("tendencia factor_de_potencia_10kv valor final vs comportamiento historico factor_de_potencia_10kv.jpg")
```

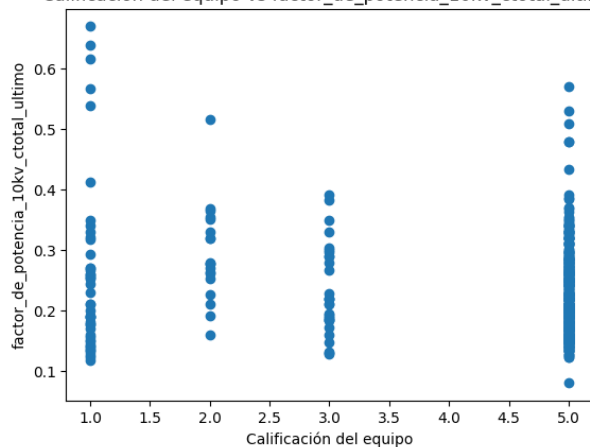
Tendencia historico factor_de_potencia_10kv vs factor_de_potencia_10kv valor final



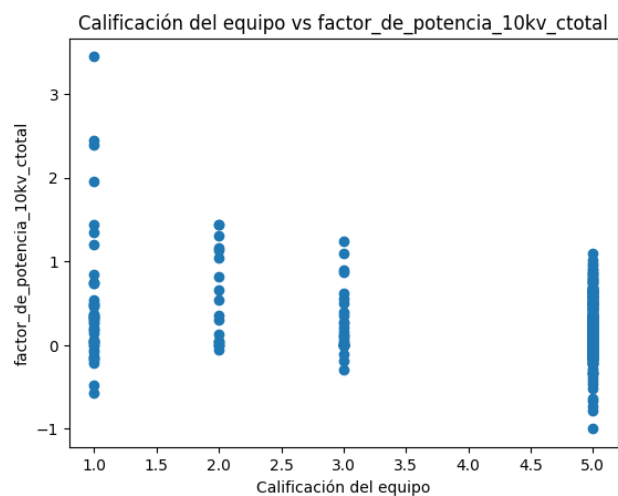
<Figure size 640x480 with 0 Axes>

```
plt.title('Calificación del equipo vs factor_de_potencia_10kv_ctotal_ultimo')
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['factor_de_potencia_10kv_ctotal_ultimo'])
plt.xlabel('Calificación del equipo')
plt.ylabel('factor_de_potencia_10kv_ctotal_ultimo')
plt.savefig("Calificación del equipo vs factor_de_potencia_10kv_ctotal_ultimo.jpg")
```

Calificación del equipo vs factor_de_potencia_10kv_ctotal_ultimo



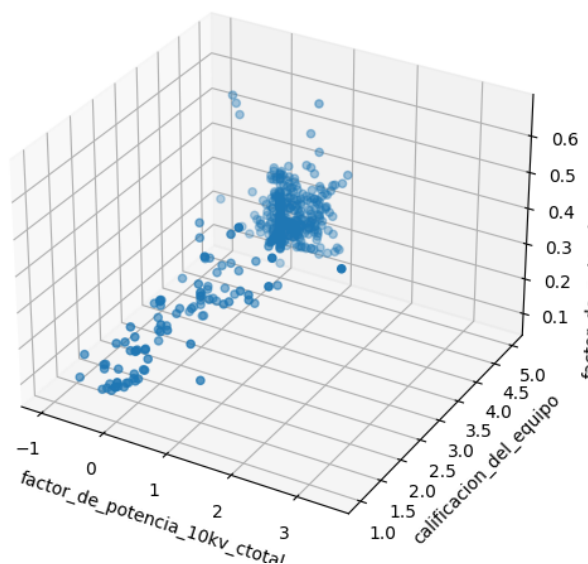
```
plt.title('Calificación del equipo vs factor_de_potencia_10kv_ctotal')
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['factor_de_potencia_10kv_ctotal'])
plt.xlabel('Calificación del equipo')
plt.ylabel('factor_de_potencia_10kv_ctotal')
plt.savefig("Calificación del equipo vs factor_de_potencia_10kv_ctotal.jpg")
```



```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
```

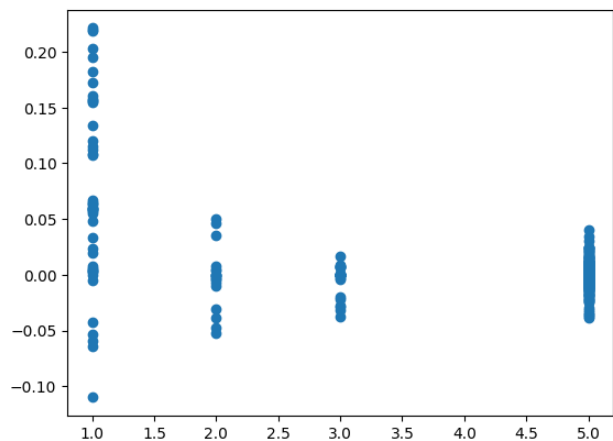
```
# Crear el scatter plot 3D
```

```
fig = plt.figure()
ax.set_title('Calificación del equipo vs factor_de_potencia_10kv_ctotal vs factor_de_potencia_10kv_ctotal_ultimo')
ax = fig.add_subplot(111, projection='3d')
fig.set_size_inches(6, 6)
ax.set_xlabel('factor_de_potencia_10kv_ctotal')
ax.set_ylabel('calificacion_del_equipo')
ax.set_zlabel('factor_de_potencia_10kv_ctotal_ultimo')
ax.scatter(df_sai_copy2['factor_de_potencia_10kv_ctotal'], df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['factor_de_potencia_10kv_ctotal_ultimo'])
plt.savefig("Calificación del equipo vs factor_de_potencia_10kv_ctotal vs factor_de_potencia_10kv_ctotal_ultimo.jpg")
```



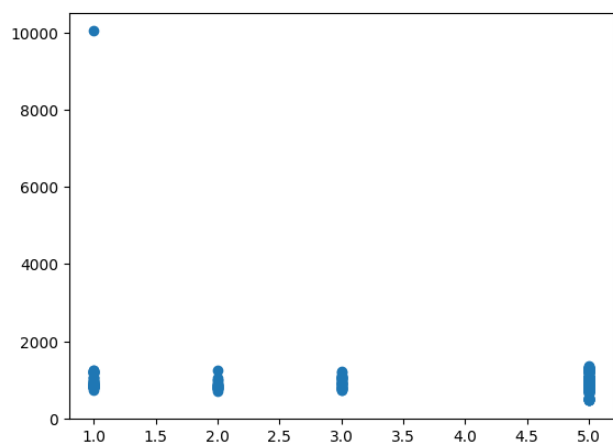
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['capacitancia_total'])
```

```
<matplotlib.collections.PathCollection at 0x7ff138519eb0>
```



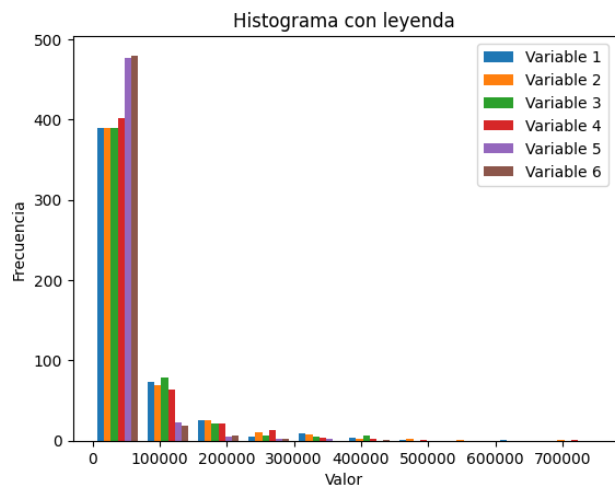
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'], df_sai_copy2['capacitancia_total_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff133f65730>



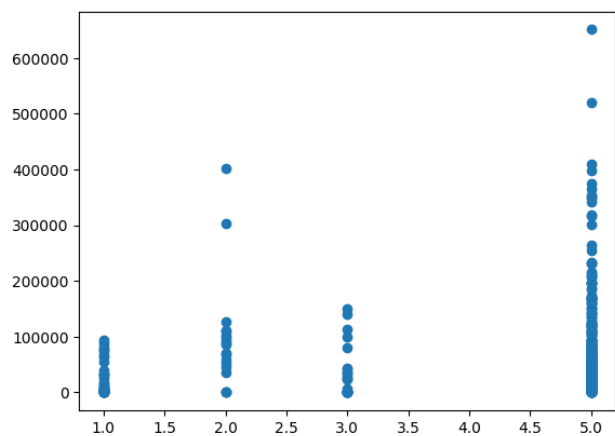
```
plt.hist([df_sai_copy2['medida_resistencia_aislamiento_nucleo_1_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_2_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_3_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_4_ultimo'], df_sai_copy2['medida_resistencia_aislamiento_nucleo_5_ultimo']], bins=10, color='blue', edgecolor='black')
plt.legend(loc='upper right')
# Agregar títulos y etiquetas de los ejes
plt.title('Histograma con leyenda')
plt.xlabel('Valor')
plt.ylabel('Frecuencia')
```

```
Text(0, 0.5, 'Frecuencia')
```

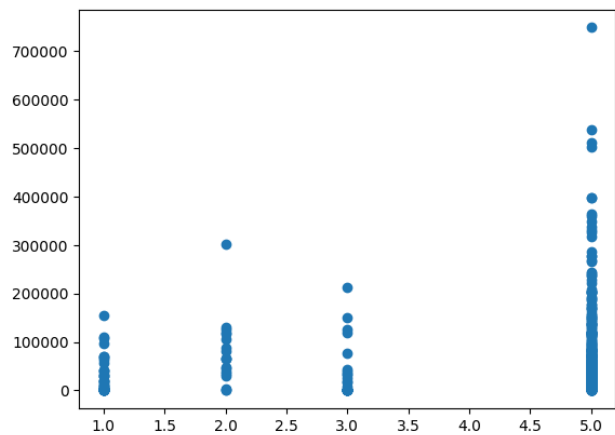
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_1_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff13821bca0>



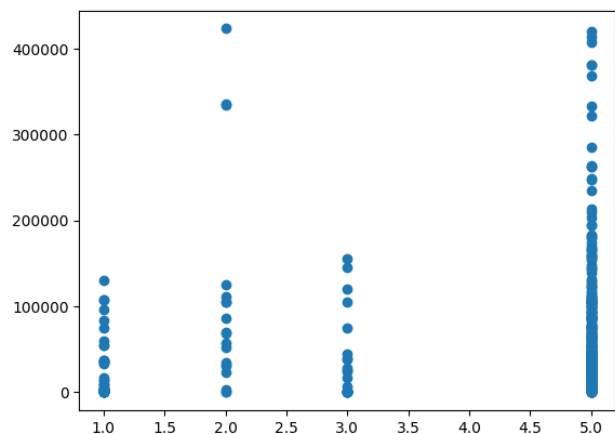
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_2_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff1383e0d90>



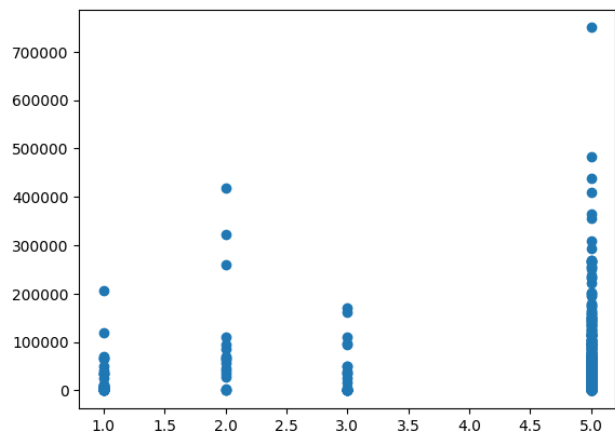
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_3_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff138435430>



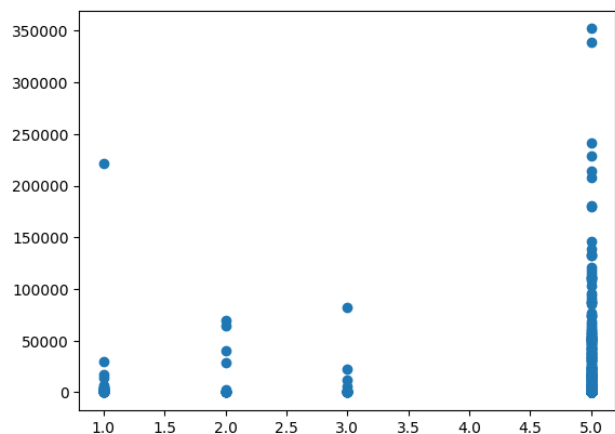
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_4_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff1382bdb50>



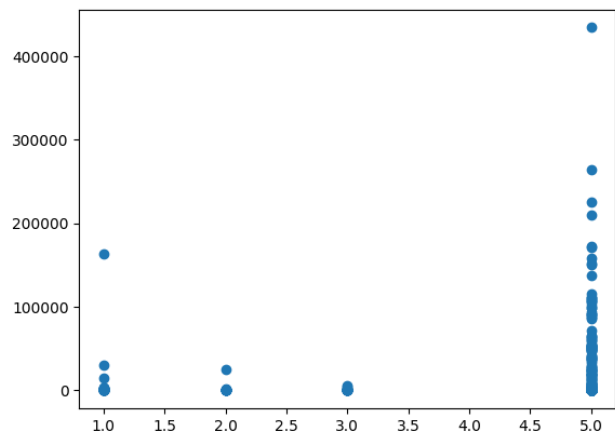
```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_5_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff138545460>



```
plt.scatter(df_sai_copy2['calificacion_del_equipo'],df_sai_copy2['medida_resistencia_aislamiento_nucleo_6_ultimo'])
```

<matplotlib.collections.PathCollection at 0x7ff138593490>



```
import matplotlib.pyplot as plt
import numpy as np

# Generar datos aleatorios para x e y
x = np.random.rand(50)
y = np.random.rand(50)

# Calcular los coeficientes de la línea de regresión
coefficients = np.polyfit(x, y, 1)

# Crear el scatter plot
plt.scatter(x, y)

# Calcular los valores de y correspondientes a los valores de x
y_fit = np.polyval(coefficients, x)

# Graficar la línea de tendencia
plt.plot(x, y_fit, 'r-')

# Agregar títulos y etiquetas de los ejes
plt.title('Scatter plot con línea de tendencia')
plt.xlabel('Eje x')
plt.ylabel('Eje y')

# Mostrar el gráfico
plt.show()
```

