Verificacion de la calidad para Viñedo de los Alpes

Objetivo

Determinar la calidad de un vino con base en parámetros fisicoquímicos históricos por medio de métodos de Machine Learning para evitar tener que contratar expertos humanos.

Aplicación de machine learning

Tarea a realizar: Estimar la calificación de calidad ya que se puede estimar de los parametros fisioquimicos. Método/algoritmo seleccionado: Regresión líneal debido a que se desea estimar una variable continua y hay variables con correlación lineal.

Importar librerias

In [152]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import KFold, GridSearchCV, train_test_split
from sklearn.preprocessing import OneHotEncoder, MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import seaborn as sns
from sklearn.metrics import mean_squared_error as mse
```

Perfilamiento y pre-procesamiento de los datos

Leer datos

In [153]:
```
datos = pd.read_csv("vinosAlpes.csv",delimiter=";")
```

Perfilamiento

In [154]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2037 entries, 0 to 2036
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   acidezTotal           2037 non-null   float64
 1   acidezVolatil         2037 non-null   float64
 2   acidoCitrico          2037 non-null   float64
 3   azucaresResiduales    2037 non-null   float64
 4   cloruros              2037 non-null   float64
 5   dioxidoLibreSulfuro   2037 non-null   float64
 6   TotalDioxidoSulfurico 2037 non-null   float64
 7   densidad              2037 non-null   float64
 8   pH                    2037 non-null   float64
 9   sulfitos              2037 non-null   float64
 10  nivelCalidad          2037 non-null   int64  
 11  grdAlcohol            2035 non-null   float64
 12  tipoVino              1842 non-null   object 
 13  calificacionCalidad   2035 non-null   float64
dtypes: float64(12), int64(1), object(1)
memory usage: 222.9+ KB
```

De los datos solo hay nulos en las columnas tipoVino, grdAlcohol y calificacionCalidad.

Todas las columnas a excepcion del tipoDeVino son numericas. A continuacion su informacion estadistica:

In [155]:
```
datos.describe()
```

Out[155]:

	acidezTotal	acidezVolatil	acidoCitrico	azucaresResiduales	cloruros	dioxidoLibreSulfuro	TotalDioxidoSulfurico	densidad	pH	sulfito
count	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.00000
mean	6.825626	0.266564	0.323201	6.277590	0.042376	34.718949	136.945508	1.585837	3.186348	0.48105
std	0.753302	0.076768	0.094378	4.867284	0.010350	15.215444	41.424123	7.292858	0.138701	0.09856
min	4.400000	0.080000	0.000000	0.700000	0.010000	3.000000	21.000000	0.990000	2.790000	0.22000
25%	6.300000	0.210000	0.270000	1.700000	0.040000	24.000000	107.000000	0.990000	3.090000	0.41000
50%	6.800000	0.260000	0.310000	5.300000	0.040000	34.000000	133.000000	0.990000	3.180000	0.47000
75%	7.300000	0.310000	0.380000	9.400000	0.050000	45.000000	166.000000	1.000000	3.280000	0.54000
max	8.800000	0.480000	0.570000	20.800000	0.070000	78.000000	253.000000	100.200000	3.560000	0.76000

Muestra de los datos:

In [156]:
```
datos.head()
```

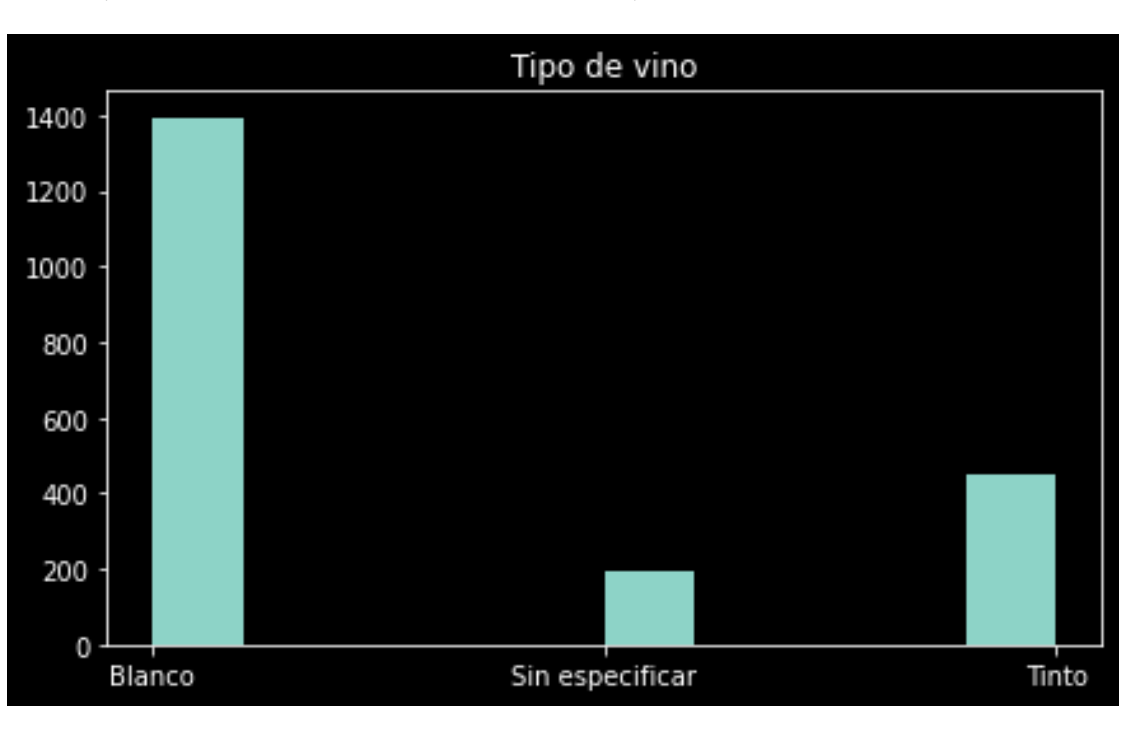
Out[156]:

	acidezTotal	acidezVolatil	acidoCitrico	azucaresResiduales	cloruros	dioxidoLibreSulfuro	TotalDioxidoSulfurico	densidad	pH	sulfitos	nivelCalidad	grdAlco
0	7.5	0.33	0.32	11.1	0.04	25.0	119.0	1.00	3.15	0.34	6	1
1	6.3	0.27	0.29	12.2	0.04	59.0	196.0	1.00	3.14	0.40	6	
2	7.0	0.30	0.51	13.6	0.05	40.0	168.0	1.00	3.07	0.52	7	
3	7.4	0.38	0.27	7.5	0.04	24.0	160.0	1.00	3.17	0.43	5	1
4	8.1	0.12	0.38	0.9	0.03	36.0	86.0	0.99	2.80	0.55	6	1

Se puede visualizar eltipo de vino de forma grafica, la mayoria son vinos blancos, seguidos de los tintos y otros sin especificar.

In [157]:
```
plt.style.use('dark_background')
fig=plt.figure(figsize=(7,4))
a = datos["tipoVino"].copy()
a[pd.isnull(a)]="Sin especificar"
plt.hist(a)
plt.title("Tipo de vino")
```

Out[157]:
```
Text(0.5, 1.0, 'Tipo de vino')
```



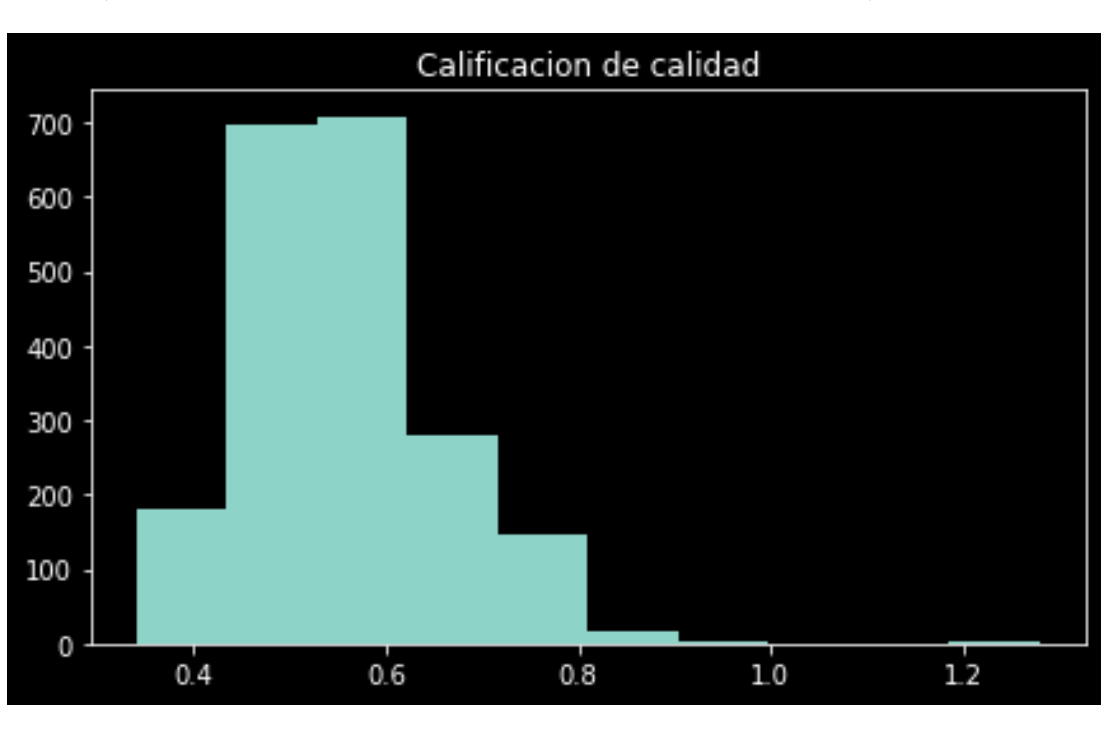
El tipo de vino no es muy relevante ya que esta informacion esta contenida en el PH, ya que tiene muchos valores nulos la eliminamos.

In [158]:
```
datos = datos.drop(["tipoVino"], axis=1)
```

Calidad: Las variables de calidad se detallan a continuacion.

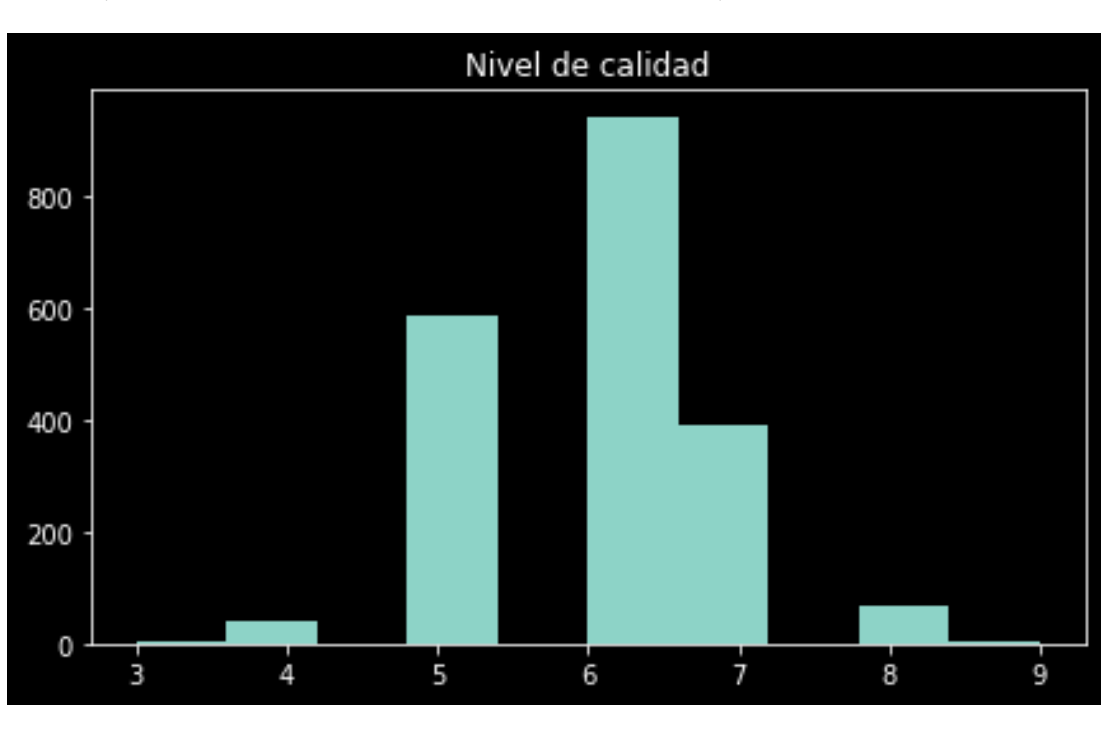
In [159]:
```
fig=plt.figure(figsize=(7,4))
plt.hist(datos["calificacionCalidad"])
plt.title("Calificacion de calidad")
```

Out[159]:
```
Text(0.5, 1.0, 'Calificacion de calidad')
```



In [160]:
```
fig=plt.figure(figsize=(7,4))
plt.hist(datos["nivelCalidad"])
plt.title("Nivel de calidad")
```

Out[160]:
```
Text(0.5, 1.0, 'Nivel de calidad')
```



La calificacion de calidad es el parametro mas interesante ya que no es categorica y de ella puede desprenderse el nivel de calidad. Así mismo, tiene una sitribucion normal. Nos deshacemos de Nivel de Calidad.

In [161]:
```
datos = datos.drop(["nivelCalidad"], axis=1)
```

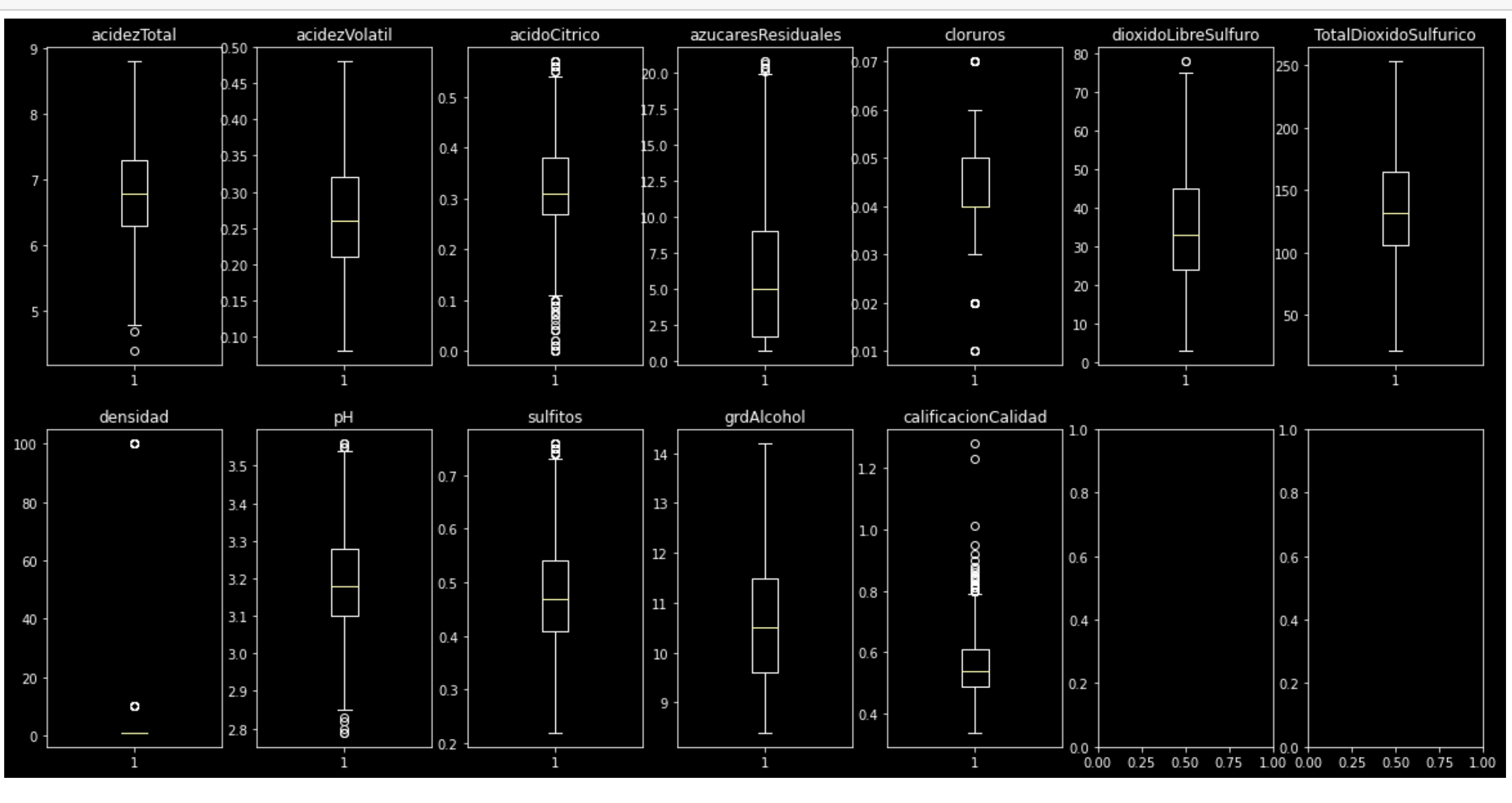
Eliminamos datos sin informacion de calificacion de calidad y repetidos.

In [162]:
```
datos = datos[datos['calificacionCalidad'].notna()]
```

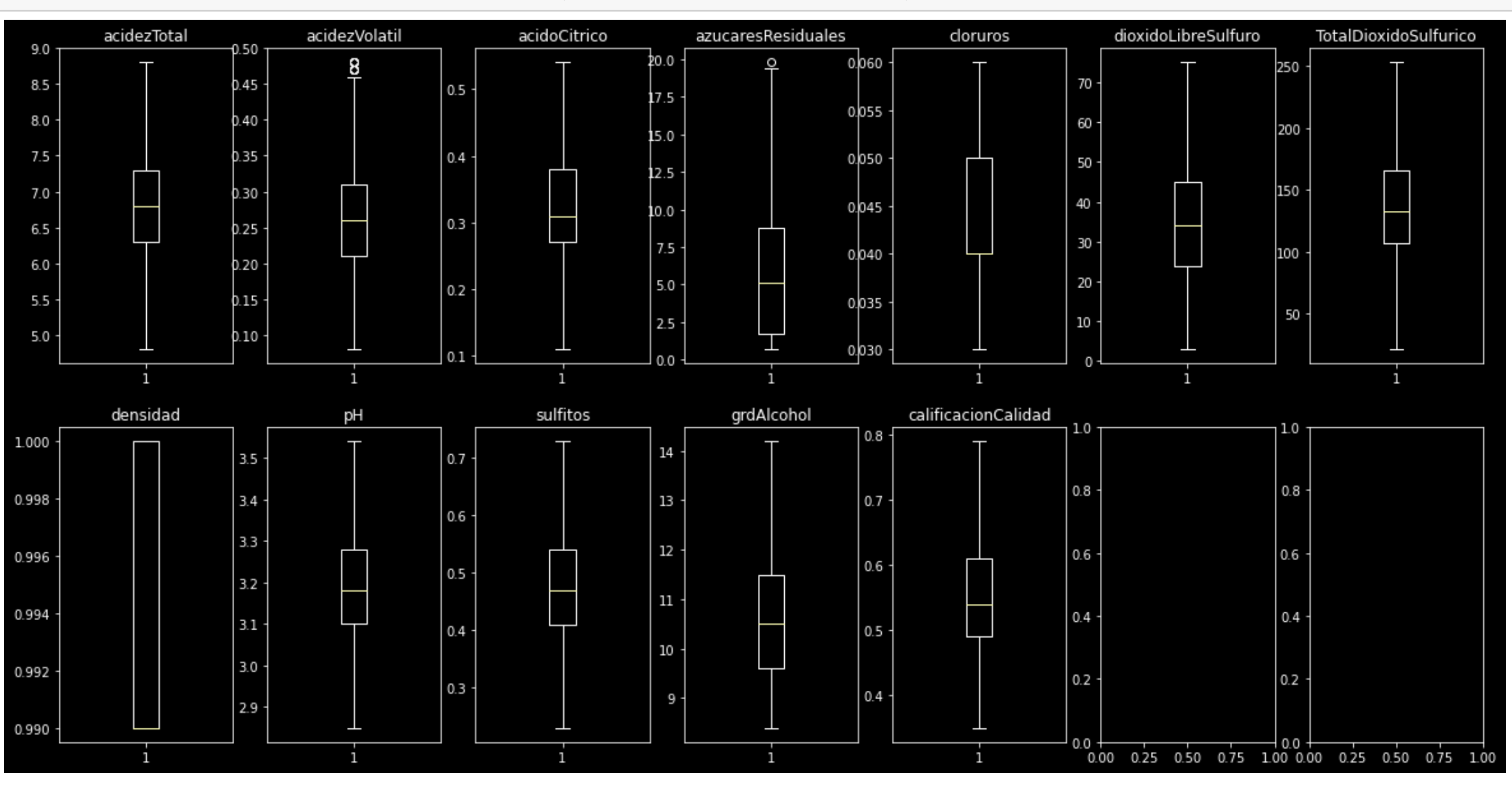
In [163]:
```
datos = datos.drop_duplicates()
```

Distribucion de los datos: Se observa que acidezVolatil cloruros, sulfitos, ph, densidad y sobre todo acidoCritico tienen datos atipicos relevantes. Por lo tanto se remueven los datpos atipicos.

In [164]:
```
datosSA = datos.copy() # datos sin anomalos
fig1, axs = plt.subplots(2,7, figsize=(20,10))
datosN = datos.select_dtypes(include='float64')
for i in range(len(datosN.columns)):
    axs[0 if i < 7 else 1,i%7].set_title(datosN.columns[i])
    di = axs[0 if i < 7 else 1,i%7].boxplot(datos[datosN.columns[i]])
    datosSA = datosSA.drop(datosSA[datosSA[datos.columns[i]]>(di["whiskers"][1].get_data()[1][1])].index)
    datosSA = datosSA.drop(datosSA[datosSA[datos.columns[i]]<(di["whiskers"][0].get_data()[1][1])].index)
```



In [165]:
```
fig1, axs = plt.subplots(2,7, figsize=(20,10))
datosN = datosSA.select_dtypes(include='float64')
for i in range(len(datosN.columns)):
    axs[0 if i < 7 else 1,i%7].set_title(datosN.columns[i])
    di = axs[0 if i < 7 else 1,i%7].boxplot(datosSA[datosN.columns[i]])
```



In [166]:
```
datosSA.shape
```

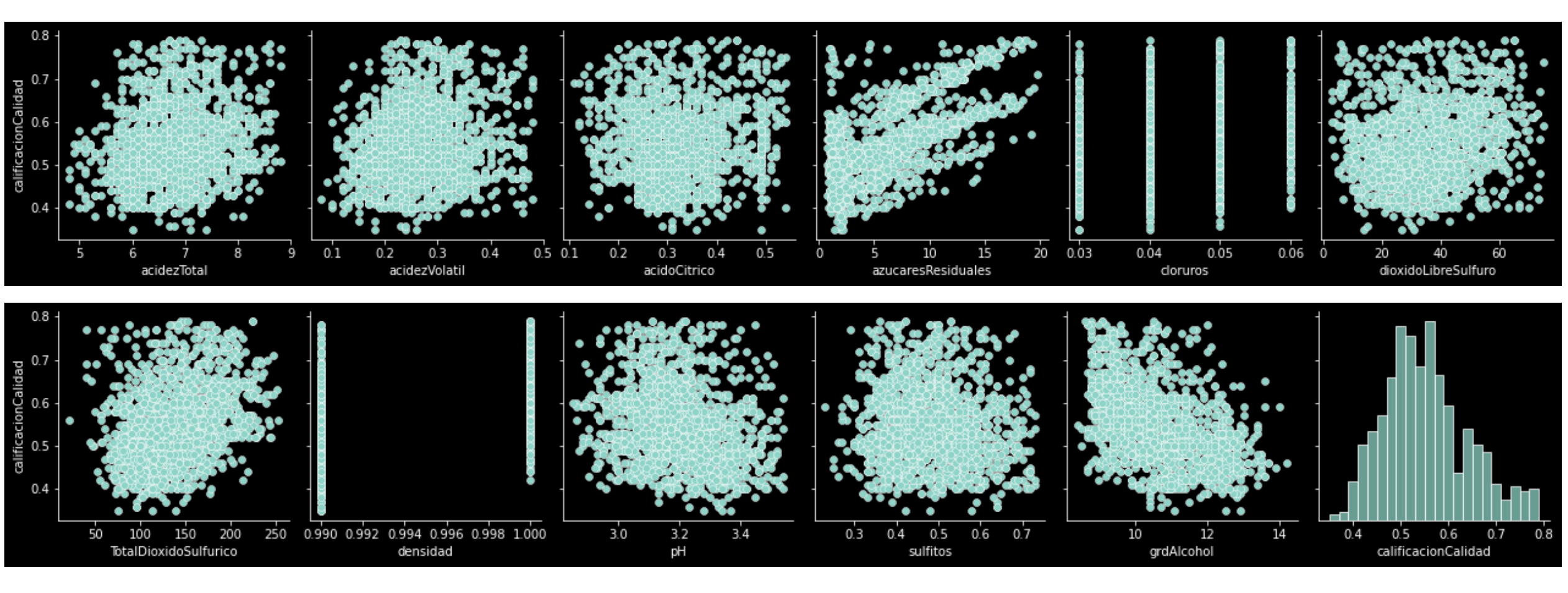
Out[166]:
```
(1618, 12)
```

Hay cerca de 400 datos atipicos que fueron eliminados.

Linealidad de los datos: Se verifica que columnas tienen linealidad y se mira su correlacion con la variable calificacionCalidad.

In [167]:
```
sns.pairplot(datosSA, height=3, y_vars = 'calificacionCalidad', x_vars = datosSA.columns[0:6], kind='scatter')
sns.pairplot(datosSA, height=3, y_vars = 'calificacionCalidad', x_vars = datosSA.columns[6:], kind='scatter')
```

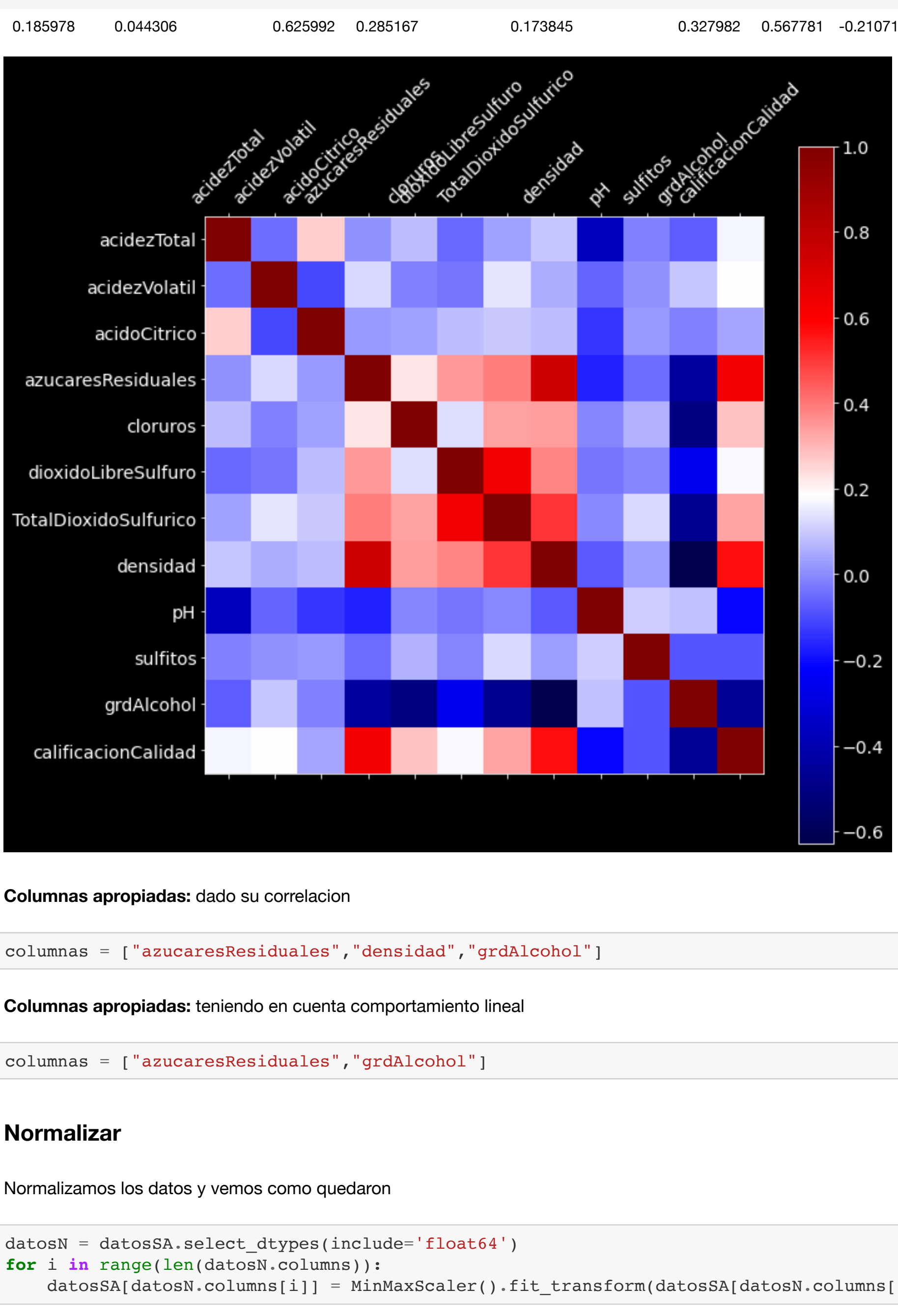
Out[167]:
```
<seaborn.axisgrid.PairGrid at 0x7fef56a1c250>
```



In [168]:
```
datosN = datosSA.select_dtypes(include='float64')
f = plt.figure(figsize=(10, 10))
plt.matshow(datosN.corr(), fignum=f.number, cmap = 'seismic')
plt.xticks(range(datosN.select_dtypes(['number']).shape[1]), datosN.select_dtypes(['number']).columns, fontsize=14, ro
tation=45)
plt.yticks(range(datosN.select_dtypes(['number']).shape[1]), datosN.select_dtypes(['number']).columns, fontsize=14)
cb = plt.colorbar()
_ = cb.ax.tick_params(labelsize=14)
datosN.corr()
```

Out[168]:

cidezVolatil	acidoCitrico	azucaresResiduales	cloruros	dioxidoLibreSulfuro	TotalDioxidoSulfurico	densidad	pH	sulfitos	grdAlcohol	calificacionCalidad
-0.046601	0.264816	0.012180	0.080460	-0.050824	0.033464	0.092549	-0.368100	-0.013179	-0.072114	0.173019
1.000000	-0.100980	0.127379	-0.017309	-0.034788	0.144719	0.055043	-0.056207	0.013544	0.095261	0.185978
-0.100980	1.000000	0.025122	0.034558	0.080816	0.097423	0.083536	-0.132372	0.020607	-0.014466	0.044306
0.127379	0.025122	1.000000	0.227158	0.347244	0.390695	0.741285	-0.167974	-0.047088	-0.443888	0.625992
-0.017309	0.034558	0.227158	1.000000	0.133204	0.335848	0.343791	-0.009252	0.064314	-0.504146	0.285167
-0.034788	0.080816	0.347244	0.133204	1.000000	0.633811	0.378531	-0.032170	-0.007972	-0.254446	0.173845
0.144719	0.097423	0.390695	0.335848	0.633811	1.000000	0.509778	-0.000079	0.126236	-0.469887	0.327982
0.055043	0.083536	0.741285	0.343791	0.378531	0.509778	1.000000	-0.078832	0.031052	-0.628481	0.567781
-0.056207	-0.132372	-0.167974	-0.009252	-0.032170	-0.000079	-0.078832	1.000000	0.107040	0.085702	-0.210719
0.013544	0.020607	-0.047088	0.064314	-0.007972	0.126236	0.031052	0.107040	1.000000	-0.083775	-0.084108
0.095261	-0.014466	-0.443888	-0.504146	-0.254446	-0.469887	-0.628481	0.085702	-0.083775	1.000000	-0.467110
0.185978	0.044306	0.625992	0.285167	0.173845	0.327982	0.567781	-0.210719	-0.084108	-0.467110	1.000000



Columnas apropiadas: dado su correlacion

In [169]:
```
columnas = ["azucaresResiduales","densidad","grdAlcohol"]
```

Columnas apropiadas: teniendo en cuenta comportamiento lineal

In [170]:
```
columnas = ["azucaresResiduales","grdAlcohol"]
```

Normalizar

Normalizamos los datos y vemos como quedaron

In [171]:
```
datosN = datosSA.select_dtypes(include='float64')
for i in range(len(datosN.columns)):
    datosSA[datosN.columns[i]] = MinMaxScaler().fit_transform(datosSA[datosN.columns[i]].to_numpy().reshape(-1, 1))
```

In [172]:
```
datos.head()
```

Out[172]:

	acidezTotal	acidezVolatil	acidoCitrico	azucaresResiduales	cloruros	dioxidoLibreSulfuro	TotalDioxidoSulfurico	densidad	pH	sulfitos	grdAlcohol	calificaci
0	7.5	0.33	0.32	11.1	0.04	25.0	119.0	1.00	3.15	0.34	10.5	
1	6.3	0.27	0.29	12.2	0.04	59.0	196.0	1.00	3.14	0.40	8.8	
2	7.0	0.30	0.51	13.6	0.05	40.0	168.0	1.00	3.07	0.52	9.6	
3	7.4	0.38	0.27	7.5	0.04	24.0	160.0	1.00	3.17	0.43	10.0	
4	8.1	0.12	0.38	0.9	0.03	36.0	86.0	0.99	2.80	0.55	12.0	

Regresion

Dividir datos

Dividimos los datos en entrenamiento y prueba

In [173]:
```
x = datosSA[columnas]
y = datosSA["calificacionCalidad"]
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42)
```

Entrenar modelo

In [174]:
```
reg = LinearRegression().fit(x_train, y_train)
```

Parametros obtenidos del modelo

Se buscan los parametros obtenidos. El intercepto no indica mucho para el negocio en este caso. Pero los coeficiientes si.

In [175]:
```
print(reg.intercept_)
reg.coef_
```

```
0.4121927360696951
```

Out[175]:
```
array([ 0.4699024 , -0.22631274])
```

Evaluar modelo

Se calcula el r^2

In [176]:
```
reg.score(x_train,y_train)
```

Out[176]:
```
0.46774794119183505
```

Se calcula el error medio cadrado

In [177]:
```
y_predicted = reg.predict(x_test)
np.sqrt(mse(y_test, y_predicted))
```

Out[177]:
```
0.16687417526334547
```

Conclusiones

Tomando como base los datos proporcionados fue posible hacer un modelo lineal que se basa en los azucares residuales y el grado de alcohol para obtener una calificacion de calidad. Esta calificacion tiene un error medio cuadrado de 0.17 -una buena valor- y un R^2 de 0.47 que aun se puede mejorar. Así mismo, el modelo permitio evidenciar que cada vez que aumenta una unidad el nivel de azucar, aumenta en media unidad la calificacion de de calidad y al aumentar el grado de alcohol en una unidad disminuye la calificacion de calidad en un quinto de unidad.

Como propuesta futura se plantea la posibilidad de hacer un arbol de clasificacion usando la variable nivelCalidad como objetivo y removiendo la variable clasificacion de calidad.