Cargar Datos

In [72]:
```
import pandas as pd
```

In [73]:
```
df = pd.read_csv('./water_potability.csv')
df.sample(10)
```

Out[73]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Tri
2686	7.945909	213.066407	16769.890546	4.745340	292.419247	478.166710	14.189856	
818	5.433466	177.828302	31421.731633	4.584134	347.097354	490.284674	16.066439	
1872	6.136907	151.784319	20561.694731	8.487856	384.156079	363.618492	13.713703	
433	8.410461	234.876524	27554.345263	5.681716	362.489560	519.031625	14.482213	
379	9.443359	73.492234	20438.224690	8.024953	315.805659	458.677231	12.538681	
1825	8.552782	217.803318	39030.603705	6.986705	373.746193	340.566245	19.084883	
3262	8.378108	198.511213	28474.202580	6.477057	319.477187	499.866994	15.389083	
776	NaN	155.864382	28224.774178	8.366723	392.582582	421.343736	18.778696	
2442	6.578681	203.408815	22374.824910	6.248929	399.617217	547.702137	12.097920	
1961	NaN	205.235194	22613.297485	6.485810	266.639384	313.009639	11.623605	

In [76]:
```
df=df.dropna()
df.sample(10)
```

Out[76]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Tri
3141	6.658742	216.564702	25172.585759	6.785521	330.517558	620.448963	19.095091	
1722	7.566517	205.396582	30823.730490	7.816636	354.175972	395.297275	12.095251	
416	6.262799	206.889748	31414.525805	4.528076	349.734662	567.027274	15.963540	
2307	9.808258	220.049574	34132.067979	9.752751	233.870327	367.044379	13.498665	
680	6.704635	230.766940	9727.761716	5.943695	223.235816	405.761571	12.826509	
1918	5.808976	157.552238	7965.207918	6.680188	262.995756	377.697283	17.401039	
1455	7.893818	203.296621	16853.676328	7.334428	339.767579	398.989500	19.318760	
2563	7.506111	188.221812	31920.584694	5.714312	334.243304	436.396995	15.220967	
956	4.713117	209.342051	20070.567792	6.591109	301.965541	354.170181	14.023834	
726	0.227499	152.530111	39028.599340	3.462492	283.693782	443.029232	13.201943	

In [77]:
```
df = pd.get_dummies(data=df, drop_first=True)
```

Selecciono las variables

In [78]:
```
explicativas = df.drop(columns='Potability')
objetivo = df.Potability
```

Entrenar modelo Arbol de Decision Clasificacion

In [79]:
```
fit()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [79], in <cell line: 1>()
----> 1 fit()

NameError: name 'fit' is not defined
```

In [80]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [81]:
```
model = DecisionTreeClassifier(max_depth=3)
```

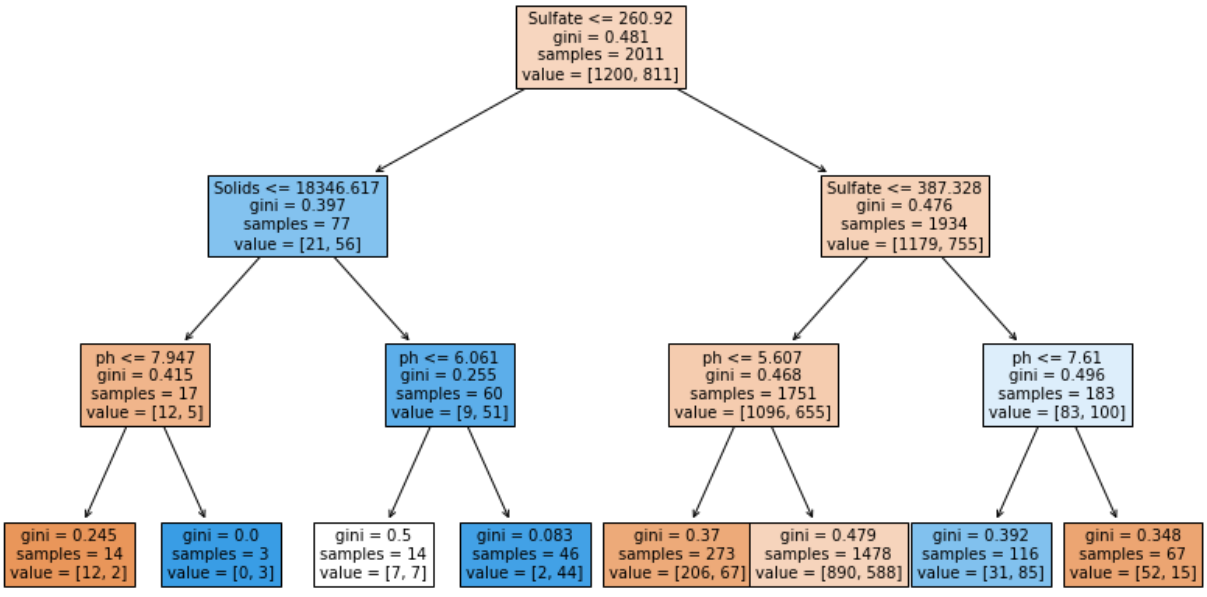
In [82]:
```
model.fit(X=explicativas,y=objetivo)
```

Out[82]:
```
DecisionTreeClassifier(max_depth=3)
```

Visualizar el Modelo

In [83]:
```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

In [84]:
```
plt.figure(figsize=(14,8))
plot_tree(decision_tree=model,feature_names=explicativas.columns,filled=True, fontsize
```



Calcular Prediccion

In [85]:
```
a = explicativas.sample()
```

In [86]:
```
a
```

Out[86]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trih
487	7.689358	221.356885	30253.851103	6.269309	320.478106	529.746529	17.973277	

In [87]:
```
model.predict_proba(a)
```

Out[87]:
```
array([[0.60216509, 0.39783491]])
```

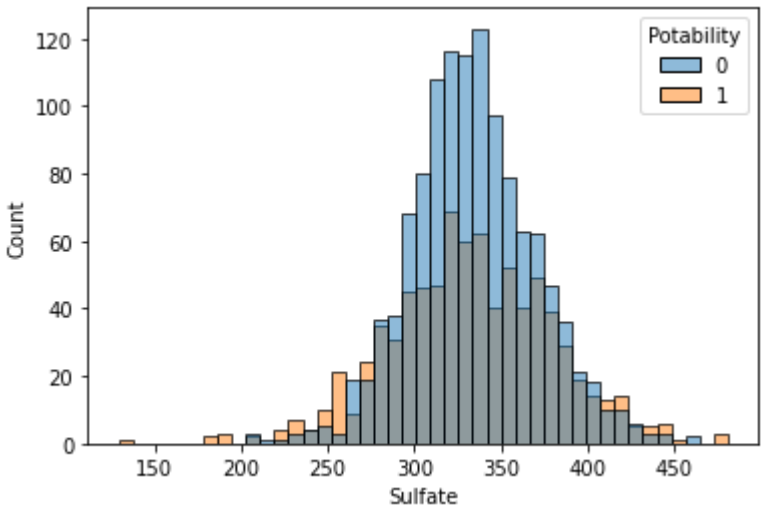
In [88]:
```
y_pred = model.predict(explicativas)
```

Interpretar Modelo

In [90]:
```
import seaborn as sns
```

In [91]:
```
sns.histplot(x=df.Sulfate, hue=df.Potability)
```

Out[91]:
```
<AxesSubplot:xlabel='Sulfate', ylabel='Count'>
```



Que tan bueno es el modelo ?

In [92]:
```
df['pred'] = y_pred
```

In [93]:
```
df.sample(10)[['Potability','pred']]
```

Out[93]:

	Potability	pred
1597	1	0
332	1	1
3016	0	0
581	0	0
3255	1	0
1890	1	0
1738	0	0
2468	0	0
2010	1	0
170	0	0

In [94]:
```
(df['Potability']==df['pred']).sum()
```

Out[94]:
```
1299
```

In [95]:
```
(df['Potability']==df['pred']).mean()
```

Out[95]:
```
0.6459472899055196
```

In [ ]:
```

```