Importamos la librerias

In [41]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from  sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import accuracy_score
```

In [42]:
```
df = pd.read_csv('./water_potability.csv')
df=df.dropna()
df.sample(10)
```

Out[42]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Tr
1110	4.736405	203.276420	28698.729965	5.271367	323.683683	508.820341	14.063140	
2243	6.014540	205.482940	23268.243081	9.035769	283.340368	371.403475	16.863668	
2694	3.676845	173.677702	22581.224488	12.653362	289.126846	599.729246	8.386407	
3128	4.959853	215.854869	9887.830755	6.954231	379.504731	527.479694	14.326638	
212	5.304143	160.968358	25299.511557	6.286043	300.500689	497.790391	15.013030	
1453	8.034012	197.609392	25204.674082	6.501097	330.552173	490.821500	15.107289	
1434	7.322502	196.104298	19534.421648	6.661157	297.841381	418.075511	17.161768	
2084	7.094831	139.906866	18975.643495	7.571951	423.045974	528.266523	18.556177	
2167	8.180755	178.079638	25623.867503	7.574082	391.918229	353.507014	19.749408	
523	8.437876	183.106625	12856.928695	7.400019	361.779682	587.022609	15.926921	

Selecciono las variables

In [43]:
```
parametros = df.drop(columns="Potability")
objetivo = df.Potability
```

Implementacion del model

Preparamos los datos de prueba

In [44]:
```
X_train, X_test, y_train,y_test =train_test_split(parametros,objetivo,test_size=0.5)
```

Escalamos los datos

In [45]:
```
escalar = StandardScaler()
```

In [46]:
```
X_train = escalar.fit_transform(X_train)
X_test = escalar.transform(X_test)
```

Defino el modelo

In [47]:

```
model = LogisticRegression()
```

Entreno el modelo

In [48]:

```
model.fit(X_train,y_train)
```

Out[48]:

```
LogisticRegression()
```

Realizo una prediccion

In [49]:

```
y_pred = model.predict(X_test)
```

Verifico la prediccion

In [50]:

```
matriz = confusion_matrix(y_test,y_pred)
print("Matriz de Confusion",matriz)
```

```
Matriz de Confusion [[590  30]
 [365  21]]
```

Precision del Modelo

In [51]:

```
precision = precision_score(y_test,y_pred)
print("Precision del modelo",precision)
```

```
Precision del modelo 0.4117647058823529
```

Exactitud del Modelo

In [52]:

```
exactitud = accuracy_score(y_test,y_pred)
print("Exactitud del modelo: ",exactitud)
```

```
Exactitud del modelo:  0.6073558648111332
```

In [ ]:

```

```