Análisis de Componentes Principales - SKLearn

In [1]:

```
import numpy as np
import pandas as pd
import plotly as py
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
init_notebook_mode(connected=True)
```

In [2]:

```
df = pd.read_csv("../../Data-Sets/datasets/iris/iris.csv")
```

In [10]:

```
X = df.iloc[:,0:4].values
y = df.iloc[:,4].values
X_std = StandardScaler().fit_transform(X) #Centrado en cero
```

In [11]:

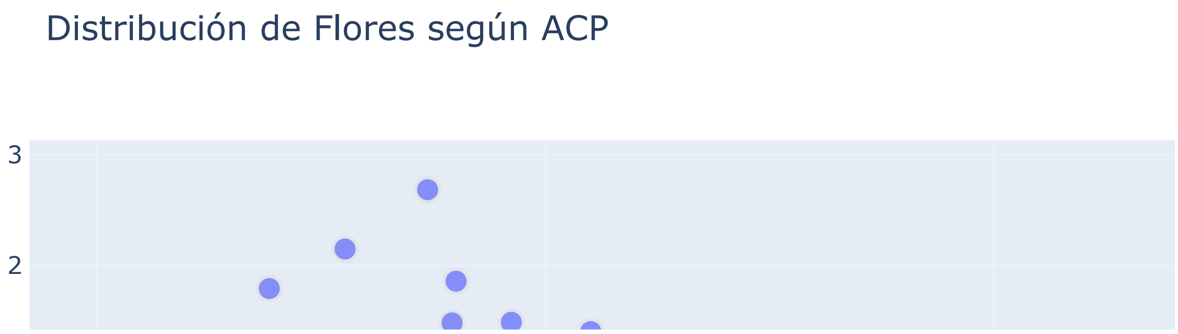
```
from sklearn.decomposition import PCA as sk_pca
```

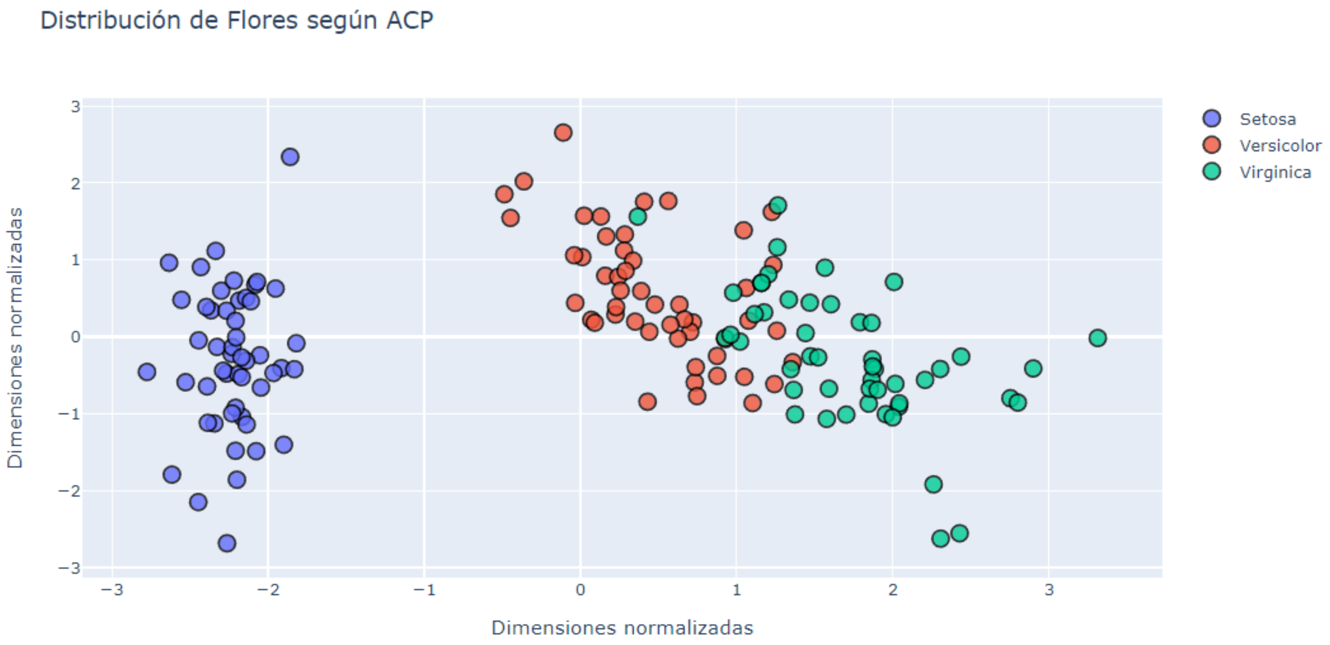
In [16]:

```
acp = sk_pca(n_components=2) # Ya conocemos el n° óptimo de componentes, que es dos
Y= acp.fit_transform(X_std) #Utiliza sigular value decomposition
```

In [26]:

```
results = []

for name in ('Setosa', 'Versicolor', 'Virginica'):
    result = go.Scatter(
                x=Y[y==name, 0],
                y=Y[y==name, 1],
                mode="markers",
                name=name,
                marker = dict(size=12, line = dict(color = 'rgb(225,225,225,0.2)', width
                )
    results.append(result)

layout = dict(showlegend=True, title='Distribución de Flores según ACP',
              xaxis= dict(title= 'CP1', ticklen= 5, zeroline= True, showline=True),
              yaxis= dict(title= 'CP2', ticklen= 5, zeroline= True, showline=True)
             )

fig = go.Figure(data = results, layout = layout)

iplot(fig)
```




Distribución de Flores según ACP
Dimensiones normalizadas
Dimensiones normalizadas
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
● Setosa
● Versicolor
● Virginica

In [ ]: