Creado por:

Isabel Maniega

In [1]:

```
# pip install seaborn
```

In [2]:

```
import pandas as pd
import numpy as np

# Gráficos
# ==============================================================
import matplotlib.pyplot as plt
import seaborn as sns
```

Ejemplo 1: Notas

In [3]:

```
df = pd.DataFrame({"notas_1": [15,16,15,17,14,14,14,10,15,25], "notas_2": [16,21,16
                   "notas_3":[17,22,15,22,14,15,16,15,24,16]})
df.head()
```

Out[3]:

	notas_1	notas_2	notas_3
0	15	16	17
1	16	21	22
2	15	16	15
3	17	16	22
4	14	13	14

Variabilidad

Desviación estándar

Calculamos la desviación estandar de los datos:

Delta Degrees of Freedom --> ddof

https://stackoverflow.com/questions/41204400/what-is-the-difference-between-numpy-var-and-statistics-variance-in-python (https://stackoverflow.com/questions/41204400/what-is-the-difference-between-numpy-var-and-statistics-variance-in-python)

In [4]:

```
std_1 = df["notas_1"].std(ddof=0)
std_1
```

Out[4]:

```
3.6124783736376886
```

In [5]:

```
std_2 = df["notas_2"].std(ddof=0)
std_2
```

Out[5]:

```
2.749545416973504
```

In [6]:

```
std_3 = df["notas_3"].std(ddof=0)
std_3
```

Out[6]:

```
3.4409301068170506
```

Varianza

In [7]:

```
varianza_1 = df["notas_1"].var(ddof=0)
varianza_1
```

Out[7]:

```
13.05
```

In [8]:

```
varianza_2 = df["notas_2"].var(ddof=0)
varianza_2
```

Out[8]:

```
7.56
```

In [9]:

```
varianza_3 = df["notas_3"].var(ddof=0)
varianza_3
```

Out[9]:

```
11.84
```

Rango Intercuartílico o IQR

Como calcular la IQR de las distintas notas:

In [11]:

```
rango_1 = df["notas_1"].max() - df["notas_1"].min()
iqr_1 = df["notas_1"].quantile(0.75) - df["notas_1"].quantile(0.25)
print(f"IQR de las notas 1: {iqr_1}, rango: {rango_1}")
```

```
IQR de las notas 1: 1.75, rango: 15
```

In [12]:

```
rango_2 = df["notas_2"].max() - df["notas_2"].min()
iqr_2 = df["notas_2"].quantile(0.75) - df["notas_2"].quantile(0.25)
print(f"IQR de las notas 2: {iqr_2}, rango: {rango_2}")
```

```
IQR de las notas 2: 3.25, rango: 9
```

In [13]:

```
rango_3 = df["notas_3"].max() - df["notas_3"].min()
iqr_3 = df["notas_3"].quantile(0.75) - df["notas_3"].quantile(0.25)
print(f"IQR de las notas 3: {iqr_3}, rango: {rango_3}")
```

```
IQR de las notas 3: 5.75, rango: 10
```

Finding Outliers

- **Notas 1:**

--> Superiores:

In [14]:

```
superiores_1 = df["notas_1"].quantile(0.75) + 1.5 * iqr_1
print(superiores_1)
```

```
18.375
```

In [15]:

```
df.notas_1
```

Out[15]:

```
0    15
1    16
2    15
3    17
4    14
5    14
6    14
7    10
8    15
9    25
Name: notas_1, dtype: int64
```

Todos los valores superiores a 18.375 son outliers, en nuestro caso es el valor 25.

--> Inferiores

In [17]:

```
inferiores_1 = df["notas_1"].quantile(0.25) - 1.5 * iqr_1
inferiores_1
```

Out[17]:

```
11.375
```

Todos los valores inferiores a 11.375 son considerados outliers, en este caso el valor 10.

- **Notas 2:**

--> Superiores:

In [18]:

```
superiores_2 = df["notas_2"].quantile(0.75) + 1.5 * iqr_2
print(superiores_2)
```

```
23.125
```

In [19]:

```
df.notas_2
```

Out[19]:

```
0    16
1    21
2    16
3    16
4    13
5    15
6    15
7    19
8    22
9    15
Name: notas_2, dtype: int64
```

--> Inferiores

In [20]:

```
inferiores_2 = df["notas_2"].quantile(0.25) - 1.5 * iqr_2
inferiores_2
```

Out[20]:

```
10.125
```

- **Notas 3:**

--> Superiores:

In [21]:

```
superiores_3 = df["notas_3"].quantile(0.75) + 1.5 * iqr_3
print(superiores_3)
```

```
29.375
```

In [22]:

```
df.notas_3
```

Out[22]:

```
0    17
1    22
2    15
3    22
4    14
5    15
6    16
7    15
8    24
9    16
Name: notas_3, dtype: int64
```

--> Inferiores

In [23]:

```
inferiores_3 = df["notas_3"].quantile(0.25) - 1.5 * iqr_3
inferiores_3
```

Out[23]:

```
6.375
```

Máximos, Mínimos, cuartiles (Q3, Q1), Mediana/Media (Q2)

- **Notas 1:**

In [25]:

```
max_1 = df["notas_1"].max()
min_1 = df["notas_1"].min()
q3_1 = df["notas_1"].quantile(0.75)
q1_1 = df["notas_1"].quantile(0.25)
mediana_1 = df["notas_1"].median()
media_1 = df["notas_1"].mean()
```

In [31]:

```
print(f"Maximo: {max_1}, Mínimo: {min_1}, Q3: {q3_1}, Q1: {q1_1}, Media: {media_1}"
```

```
Maximo: 25, Mínimo: 10, Q3: 15.75, Q1: 14.0, Media: 15.5
```

- **Notas 2:**

In [26]:

```
max_2 = df["notas_2"].max()
min_2 = df["notas_2"].min()
q3_2 = df["notas_2"].quantile(0.75)
q1_2 = df["notas_2"].quantile(0.25)
mediana_2 = df["notas_2"].median()
media_2 = df["notas_2"].mean()
```

In [33]:

```
print(f"Maximo: {max_2}, Mínimo: {min_2}, Q3: {q3_2}, Q1: {q1_2}, Media: {media_2}"
```

```
Maximo: 22, Mínimo: 13, Q3: 18.25, Q1: 15.0, Media: 16.8
```

- **Notas 3:**

In [27]:

```
max_3 = df["notas_3"].max()
min_3 = df["notas_3"].min()
q3_3 = df["notas_3"].quantile(0.75)
q1_3 = df["notas_3"].quantile(0.25)
mediana_3 = df["notas_3"].median()
media_3 = df["notas_3"].mean()
```

In [34]:

```
print(f"Maximo: {max_3}, Mínimo: {min_3}, Q3: {q3_3}, Q1: {q1_3}, Media: {media_3}"
```

```
Maximo: 24, Mínimo: 14, Q3: 20.75, Q1: 15.0, Media: 17.6
```

In [32]:

```
df.describe()
```

Out[32]:

	notas_1	notas_2	notas_3
count	10.000000	10.000000	10.000000
mean	15.500000	16.800000	17.600000
std	3.807887	2.898275	3.627059
min	10.000000	13.000000	14.000000
25%	14.000000	15.000000	15.000000
50%	15.000000	16.000000	16.000000
75%	15.750000	18.250000	20.750000
max	25.000000	22.000000	24.000000

Resultados notas 1:

In [28]:

```
print(f'Desviación estándar: {std_1}, Varianza: {varianza_1}, Rango: {rango_1}, \nI
 Outlier Sup: {superiores_1}, Outlier Inf: {inferiores_1}')
```

```
Desviación estándar: 3.6124783736376886, Varianza: 13.05, Rango: 15,
IQR: 1.75 Outlier Sup: 18.375, Outlier Inf: 11.375
```

Resultados notas 2:

In [29]:

```
print(f'Desviación estándar: {std_2}, Varianza: {varianza_2}, Rango: {rango_2}, \nI
 Outlier Sup: {superiores_2}, Outlier Inf: {inferiores_2}')
```

```
Desviación estándar: 2.749545416973504, Varianza: 7.56, Rango: 9,
IQR: 3.25 Outlier Sup: 23.125, Outlier Inf: 10.125
```

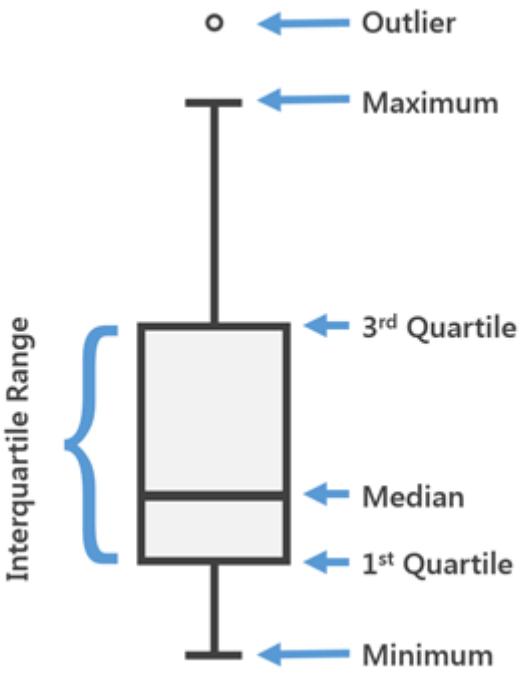
Resultados notas 3:

In [30]:

```
print(f'Desviación estándar: {std_3}, Varianza: {varianza_3}, Rango: {rango_3}, \nI
 Outlier Sup: {superiores_3}, Outlier Inf: {inferiores_3}')
```

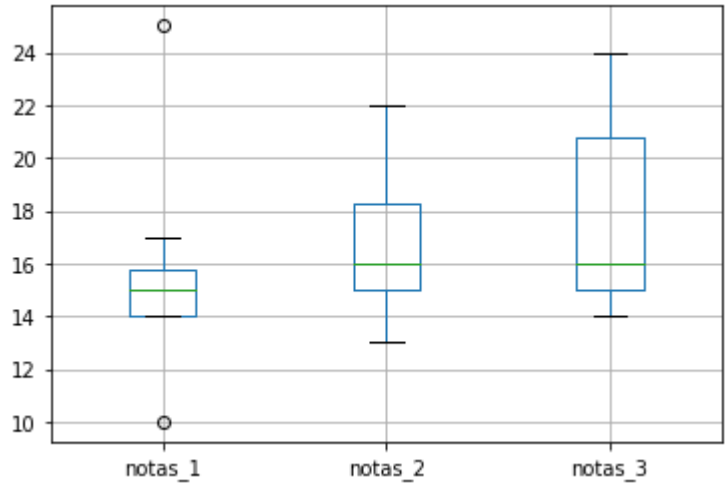
```
Desviación estándar: 3.4409301068170506, Varianza: 11.84, Rango: 10,
IQR: 5.75 Outlier Sup: 29.375, Outlier Inf: 6.375
```

Diagrama de cajas



In [36]:

```
# usando pandas:
boxplot = df.boxplot(column=["notas_1", "notas_2", "notas_3"])
```

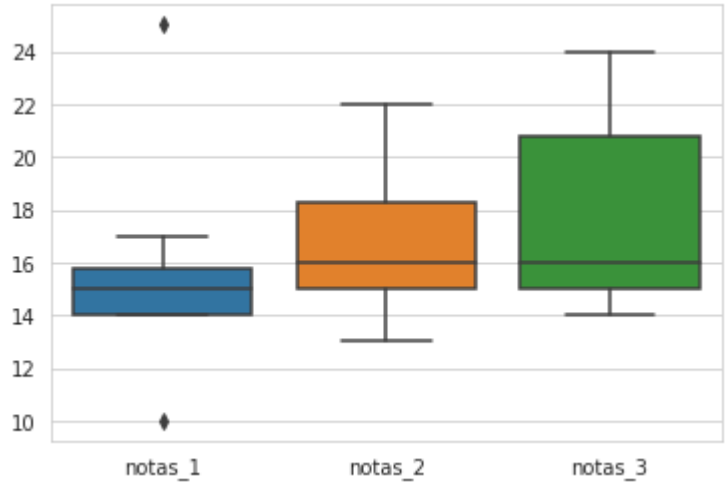


In [37]:

```
# seaborn:
sns.set_style("whitegrid")
sns.boxplot(data=df)
```

Out[37]:

```
<AxesSubplot:>
```



Ejemplo 2: Beneficios de una empresa

In [3]:

```
df_2 = pd.DataFrame({"Beneficios Enero ($)": [2500,2650,2740,2500],
                     "Beneficios Febrero ($)": [3000,3225,3000,3100],
                     "Beneficios Marzo ($)": [2900,2700,3400,2700]})
df_2.head()
```

Out[3]:

	Beneficios Enero ($)	Beneficios Febrero ($)	Beneficios Marzo ($)
0	2500	3000	2900
1	2650	3225	2700
2	2740	3000	3400
3	2500	3100	2700

Beneficios Enero

In [42]:

```
std_enero = df_2["Beneficios Enero ($)"].std(ddof=0)
varianza_enero = df_2["Beneficios Enero ($)"].var(ddof=0)
rango_enero = df_2["Beneficios Enero ($)"].max() - df_2["Beneficios Enero ($)"].min
iqr_enero = df_2["Beneficios Enero ($)"].quantile(0.75) - df_2["Beneficios Enero ($
superior_enero = df_2["Beneficios Enero ($)"].quantile(0.75) + 1.5 * iqr_enero
inferior_enero = df_2["Beneficios Enero ($)"].quantile(0.25) - 1.5 * iqr_enero
max_enero = df_2["Beneficios Enero ($)"].max()
min_enero = df_2["Beneficios Enero ($)"].min()
q3_enero = df_2["Beneficios Enero ($)"].quantile(0.75)
q1_enero = df_2["Beneficios Enero ($)"].quantile(0.25)
mediana_enero = df_2["Beneficios Enero ($)"].median()

print(f"IQR: {iqr_enero}, Outlier Sup: {superior_enero}, Outilier Inf: {inferior_en
Q1: {q1_enero}, Q3: {q3_enero}, Mediana: {mediana_enero}")
```

```
IQR: 172.5, Outlier Sup: 2931.25, Outilier Inf: 2241.25, Q1: 2500.0, Q
3: 2672.5, Mediana: 2575.0
```

Beneficios Febrero

In [44]:

```
std_f = df_2["Beneficios Febrero ($)"].std(ddof=0)
varianza_f = df_2["Beneficios Febrero ($)"].var(ddof=0)
rango_f = df_2["Beneficios Febrero ($)"].max() - df_2["Beneficios Febrero ($)"].min
iqr_f = df_2["Beneficios Febrero ($)"].quantile(0.75) - df_2["Beneficios Febrero ($
superior_f = df_2["Beneficios Febrero ($)"].quantile(0.75) + 1.5 * iqr_f
inferior_f = df_2["Beneficios Febrero ($)"].quantile(0.25) - 1.5 * iqr_f
max_f = df_2["Beneficios Febrero ($)"].max()
min_f = df_2["Beneficios Febrero ($)"].min()
q3_f = df_2["Beneficios Febrero ($)"].quantile(0.75)
q1_f = df_2["Beneficios Febrero ($)"].quantile(0.25)
mediana_f = df_2["Beneficios Febrero ($)"].median()

print(f"IQR: {iqr_f}, Outlier Sup: {superior_f}, Outilier Inf: {inferior_f}, \
Q1: {q1_f}, Q3: {q3_f}, Mediana: {mediana_f}")
```

```
IQR: 131.25, Outlier Sup: 3328.125, Outilier Inf: 2803.125, Q1: 3000.
0, Q3: 3131.25, Mediana: 3050.0
```

Beneficios Marzo

In [46]:

```
std_m = df_2["Beneficios Marzo ($)"].std(ddof=0)
varianza_m = df_2["Beneficios Marzo ($)"].var(ddof=0)
rango_m = df_2["Beneficios Marzo ($)"].max() - df_2["Beneficios Marzo ($)"].min()
iqr_m = df_2["Beneficios Marzo ($)"].quantile(0.75) - df_2["Beneficios Marzo ($)"].
superior_m = df_2["Beneficios Marzo ($)"].quantile(0.75) + 1.5 * iqr_m
inferior_m = df_2["Beneficios Marzo ($)"].quantile(0.25) - 1.5 * iqr_m
max_m = df_2["Beneficios Marzo ($)"].max()
min_m = df_2["Beneficios Marzo ($)"].min()
q3_m = df_2["Beneficios Marzo ($)"].quantile(0.75)
q1_m = df_2["Beneficios Marzo ($)"].quantile(0.25)
mediana_m = df_2["Beneficios Marzo ($)"].median()

print(f"IQR: {iqr_m}, Outlier Sup: {superior_m}, Outilier Inf: {inferior_m}, \
Q1: {q1_m}, Q3: {q3_m}, Mediana: {mediana_m}")
```

```
IQR: 325.0, Outlier Sup: 3512.5, Outilier Inf: 2212.5, Q1: 2700.0, Q3:
3025.0, Mediana: 2800.0
```

In [47]:

```
df_2.describe()
```

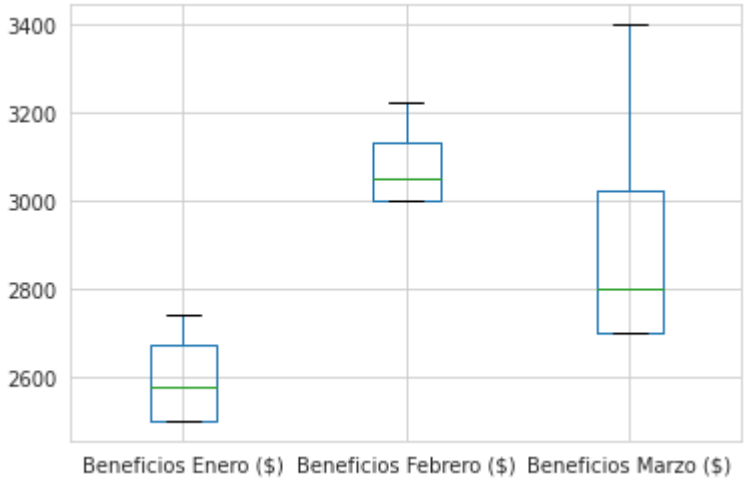
Out[47]:

	Beneficios Enero ($)	Beneficios Febrero ($)	Beneficios Marzo ($)
count	4.000000	4.000000	4.000000
mean	2597.500000	3081.250000	2925.000000
std	118.427193	106.800047	330.403793
min	2500.000000	3000.000000	2700.000000
25%	2500.000000	3000.000000	2700.000000
50%	2575.000000	3050.000000	2800.000000
75%	2672.500000	3131.250000	3025.000000
max	2740.000000	3225.000000	3400.000000

Diagrama de cajas

In [48]:

```
boxplot = df_2.boxplot(column=["Beneficios Enero ($)", "Beneficios Febrero ($)", "B
```

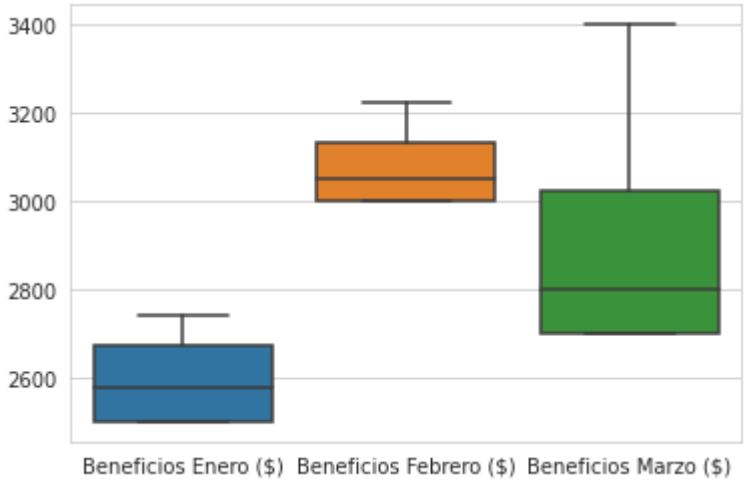


In [49]:

```
sns.set_style("whitegrid")
sns.boxplot(data=df_2)
```

Out[49]:

<AxesSubplot:>



No presenta outliers

Creado por:

Isabel Maniega