

## Actividad 3

Física Computacional 1  
Corral Valdez Jesus Giovanni  
Departamento de Física  
Universidad de Sonora

## 1 Introducción

Durante esta segunda actividad, se analizará datos parecidos a los de la Actividad 1, pero ahora son números recolectados por la propia Universidad de Sonora en el manglar de El Sargento



También se probará la herramienta de Plotly para los gráficos necesarios



## 2 Desarrollo

1. Se importó nuestra nueva herramienta y demás librerías que se utilizarán.

```
In [1]: import plotly
        plotly.__version__

Out[1]: '3.2.1rc1'

In [2]: import plotly.plotly as py
        import plotly.graph_objs as go
        import plotly.figure_factory as FF

        import numpy as np
        import pandas as pd

        import seaborn as sns
        %matplotlib inline
```

2. Sargento cuenta con dos sensores, el cual uno lo llamaremos "Canal" y el otro "Estación". Al importar sus csv de datos, le asignaremos nombre a sus columnas.

Se dividió cada una de estas listas en dos periodos, de un día y una semana

```
In [3]: dfc = pd.read_csv("canal.csv", sep=',', skiprows=2)
        dfc.columns = ['No', 'DD/MM/YY', 'AbsPresKpa', 'Temp°C', 'WaterLevelMeters']
        dfcpt2 = pd.read_csv("canal2.csv", sep=',', skiprows=2)
        dfcpt2.columns = ['No', 'DD/MM/YY', 'AbsPresKpa', 'Temp°C', 'WaterLevelMeters']

        dfe = pd.read_csv("estacion.csv", sep=',', skiprows=2)
        dfe.columns = ['No', 'DD/MM/YY', 'AbsPresKpa', 'Temp°C', 'WaterLevelMeters']
        dfcpt2 = pd.read_csv("estacion2.csv", sep=',', skiprows=2)
        dfcpt2.columns = ['No', 'DD/MM/YY', 'AbsPresKpa', 'Temp°C', 'WaterLevelMeters']

        dfs = pd.read_csv("salinidad.csv", sep=',', skiprows=2)
        dfs.columns = ['No', 'DD/MM/YY', 'CondLow_S/cm', 'CondHigh_S/cm', 'Temp°C', 'SpecificCond_S/cm', 'Salinity_ppt']

        dfc7 = dfc[(dfc['DD/MM/YY'] > '08/19/2018 23:50:39') & (dfc['DD/MM/YY'] <= '08/26/2018 23:46:50')]
        dfc1 = dfc[(dfc['DD/MM/YY'] > '08/24/2018 23:58:39') & (dfc['DD/MM/YY'] <= '08/25/2018 23:45:14')]
        dfe7 = dfe[(dfe['DD/MM/YY'] > '08/19/2018 23:45:00') & (dfe['DD/MM/YY'] <= '08/26/2018 23:45:00')]
        dfe1 = dfe[(dfe['DD/MM/YY'] > '08/24/2018 23:45:00') & (dfe['DD/MM/YY'] <= '08/25/2018 23:45:00')]
        dfs7 = dfs[(dfs['DD/MM/YY'] > '08/19/2018 23:45:00') & (dfs['DD/MM/YY'] <= '08/26/2018 23:45:00')]
        dfs1 = dfs[(dfs['DD/MM/YY'] > '08/24/2018 23:45:00') & (dfs['DD/MM/YY'] <= '08/25/2018 23:45:00')]
```

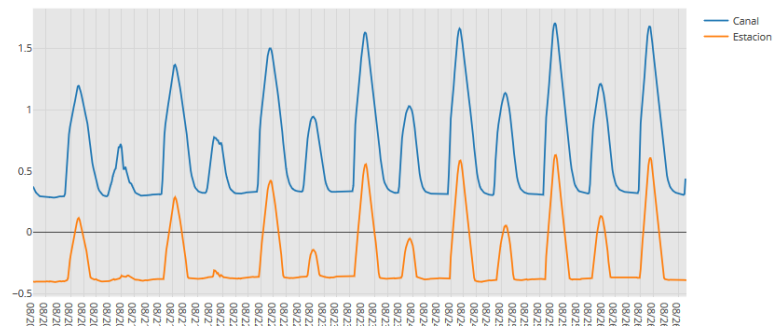
para hacer mas sencilla su comparación y además se agregó datos sobre la salinidad de la zona de la Estación.

3. Nuestras primera gráficas en Plotly serán una sencilla comparación del nivel del agua de la zona de la estación y de la zona del canal, respecto a un día y una semana.

```
In [5]: trace1 = go.Scatter(  
        x=dfe7['DD/MM/YY'], y=dfc7['WaterLevelMeters'],  
        mode='lines', name='Canal'  
    )  
    trace2 = go.Scatter(  
        x=dfe7['DD/MM/YY'], y=dfe7['WaterLevelMeters'],  
        mode='lines', name='Estacion'  
    )  
    layout = go.Layout(title='Nivel del mar en el canal y la estacion 08/20/18 - 08/26/18 ',  
        plot_bgcolor='rgb(230, 230,230)')  
    fig = go.Figure(data=[trace1,trace2], layout=layout)  
    py.iplot(fig, filename='simple-plot-from-csv')
```

Out[5]:

Nivel del mar en el canal y la estacion 08/20/18 - 08/26/18



```

In [6]: trace1 = go.Scatter(
        x=dfel['DD/MM/YY'], y=dfel['WaterLevelMeters'],
        mode='lines', name='Canal'
    )

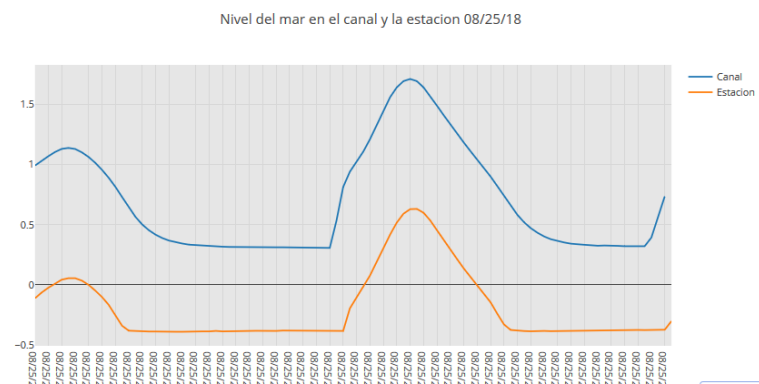
    trace2 = go.Scatter(
        x=dfel['DD/MM/YY'], y=dfel['WaterLevelMeters'],
        mode='lines', name='Estacion'
    )

    layout = go.Layout(title='Nivel del mar en el canal y la estacion 08/25/18 ',
        plot_bgcolor='rgb(230, 230,230)')

    fig = go.Figure(data=[trace1, trace2], layout=layout)
    py.iplot(fig, filename='simple-plot-from-csv')

```

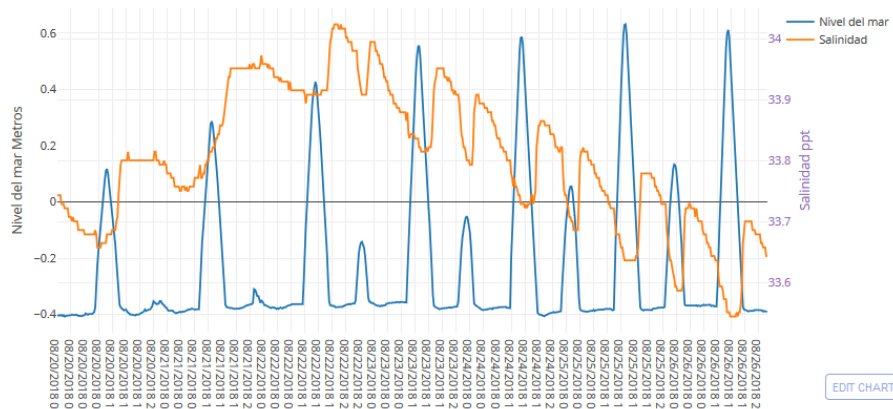
Out [6]:



4. Contamos con datos de la salinidad en la estación así que la graficaremos respecto a al nivel del mar en el mismo tiempo para encontrar si existe alguna relación.

```
In [7]: trace1 = go.Scatter(
        x=dfe7['DD/MM/YY'],
        y=dfe7['WaterLevelMeters'],
        name='Nivel del mar'
    )
    trace2 = go.Scatter(
        x=dfe7['DD/MM/YY'],
        y=dfe7['Salinity_ppt'],
        name='Salinidad',
        yaxis='y2'
    )
    data = [trace1, trace2]
    layout = go.Layout(
        title='Salinidad y Nivel del mar en la estacion 08/20/18 - 08/26/18 ',
        yaxis=dict(
            title='Nivel del mar Metros'
        ),
        yaxis2=dict(
            title='Salinidad ppt',
            titlefont=dict(
                color='rgb(148, 103, 189)'
            ),
            tickfont=dict(
                color='rgb(148, 103, 189)'
            ),
            overlaying='y',
            side='right'
        )
    )
    fig = go.Figure(data=data, layout=layout)
    py.iplot(fig, filename='simple-plot-from-csv')
```

Salinidad y Nivel del mar en la estacion 08/20/18 - 08/26/18

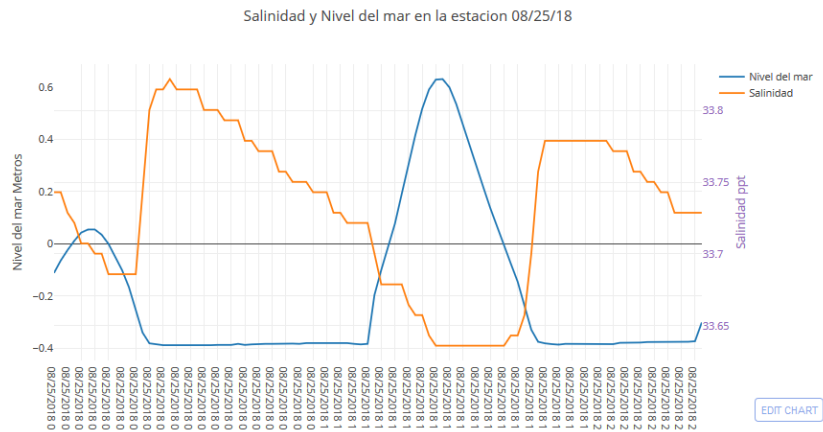


```

In [8]: trace1 = go.Scatter(
        x=dfel['DD/MM/YY'],
        y=dfel['WaterLevelMeters'],
        name='Nivel del mar'
    )
    trace2 = go.Scatter(
        x=dfel['DD/MM/YY'],
        y=dfel['Salinity_ppt'],
        name='Salinidad',
        yaxis='y2'
    )
    data = [trace1, trace2]
    layout = go.Layout(
        title='Salinidad y Nivel del mar en la estacion 08/25/18 ',
        yaxis=dict(
            title='Nivel del mar Metros'
        ),
        yaxis2=dict(
            title='Salinidad ppt',
            titlefont=dict(
                color='rgb(148, 103, 189)'
            ),
            tickfont=dict(
                color='rgb(148, 103, 189)'
            ),
            overlaying='y',
            side='right'
        )
    )
    fig = go.Figure(data=data, layout=layout)
    py.iplot(fig, filename='simple-plot-from-csv')

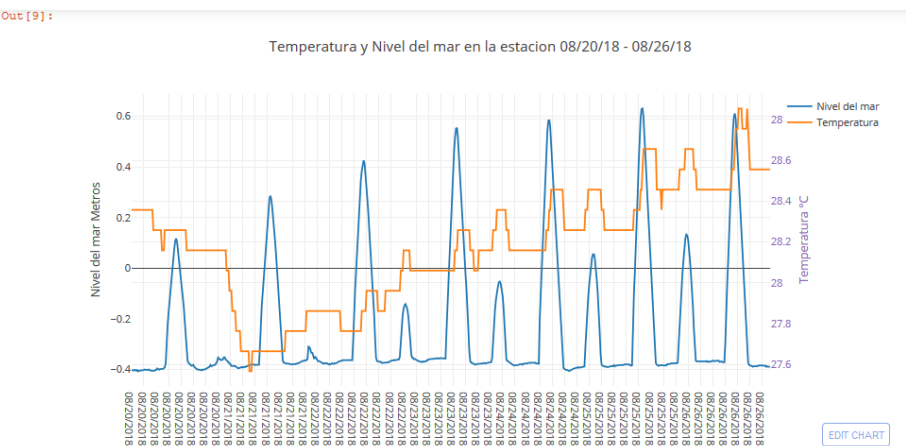
```

Out[8]:



5. También se graficaron los datos de la temperatura de la estación.

```
In [9]:
trace1 = go.Scatter(
    x=dfe7['DD/MM/YY'],
    y=dfe7['WaterLevelMeters'],
    name='Nivel del mar'
)
trace2 = go.Scatter(
    x=dfe7['DD/MM/YY'],
    y=dfe7['Temp°C'],
    name='Temperatura',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='Temperatura y Nivel del mar en la estacion 08/20/18 - 08/26/18 ',
    yaxis=dict(
        title='Nivel del mar Metros'
    ),
    yaxis2=dict(
        title='Temperatura °C',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='simple-plot-from-csv')
```

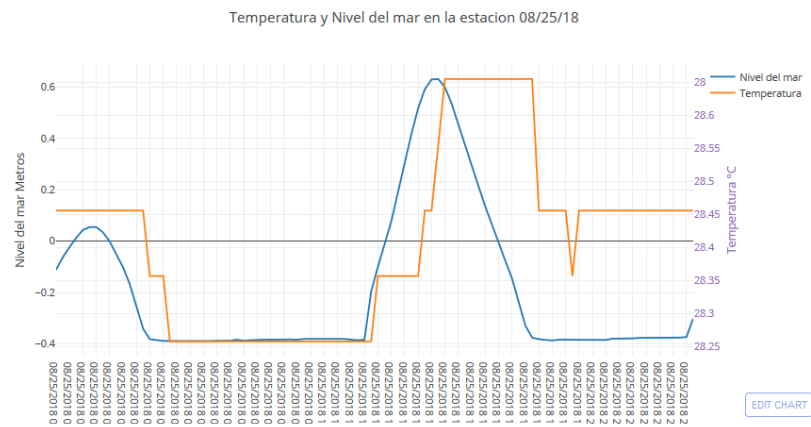




In [10]:

```
trace1 = go.Scatter(
    x=dfel['DD/MM/YY'],
    y=dfel['WaterLevelMeters'],
    name='Nivel del mar'
)
trace2 = go.Scatter(
    x=dfel['DD/MM/YY'],
    y=dfel['Temp°C'],
    name='Temperatura',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='Temperatura y Nivel del mar en la estacion 08/25/18 ',
    yaxis=dict(
        title='Nivel del mar Metros'
    ),
    yaxis2=dict(
        title='Temperatura °C',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='simple-plot-from-csv')
```

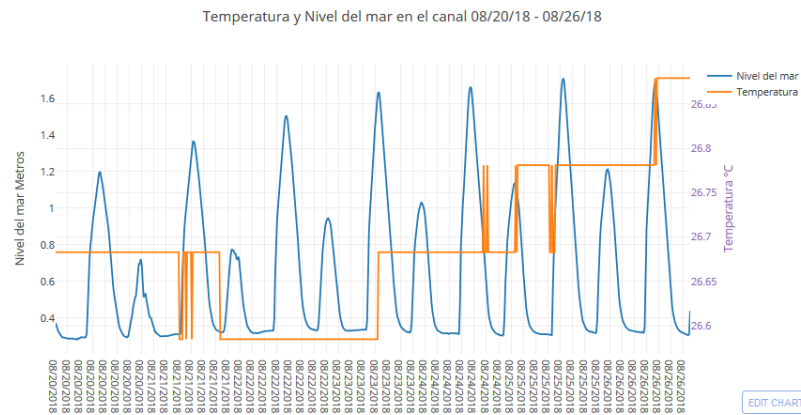
Out [10]:



6.Y ahora lo mismo con la temperatura del canal.

```
In [11]:
trace1 = go.Scatter(
    x=dfc7['DD/MM/YY'],
    y=dfc7['WaterLevelMeters'],
    name='Nivel del mar'
)
trace2 = go.Scatter(
    x=dfc7['DD/MM/YY'],
    y=dfc7['Temp°C'],
    name='Temperatura',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='Temperatura y Nivel del mar en el canal 08/20/18 - 08/26/18 ',
    yaxis=dict(
        title='Nivel del mar Metros'
    ),
    yaxis2=dict(
        title='Temperatura °C',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='simple-plot-from-csv')
```

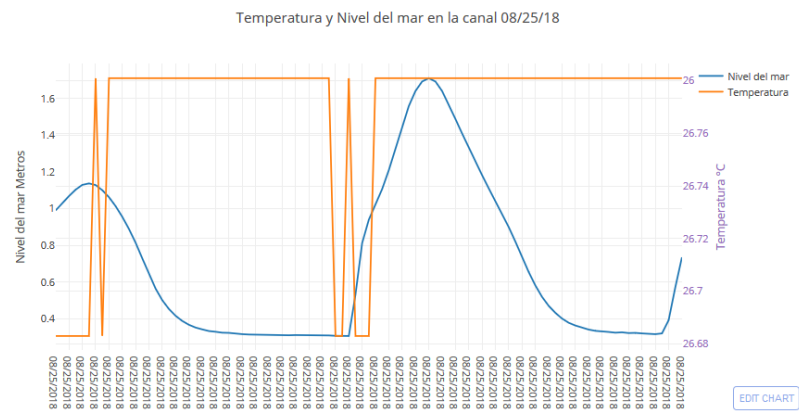
Out[11]:



In [12]:

```
trace1 = go.Scatter(
    x=dfel['DD/MM/YY'],
    y=dfcl['WaterLevelMeters'],
    name='Nivel del mar'
)
trace2 = go.Scatter(
    x=dfel['DD/MM/YY'],
    y=dfcl['Temp°C'],
    name='Temperatura',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='Temperatura y Nivel del mar en la canal 08/25/18 ',
    yaxis=dict(
        title='Nivel del mar Metros'
    ),
    yaxis2=dict(
        title='Temperatura °C',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='simple-plot-from-csv')
```

Out[12]:



7. A veces se puede obviar una correlación con las graficas, pero como nuestro objetivo es saber con mas certeza esto, se realiza una tabla de correlación entre las tablas de datos de la estación.

Y con estos datos es posible hacer un mapa de calor y encontrar las correlaciones

```
In [13]: result = pd.concat([dfel, dfs1], axis=1).corr()
result = result.drop('CondLow_S/cm', 1)
result = result.drop('CondLow_S/cm', 0)
result = result.drop('CondHigh_S/cm', 1)
result = result.drop('CondHigh_S/cm', 0)
result = result.drop('No', 0)
result = result.drop('No', 1)
```

```
result
```

```
In [14]: result
```

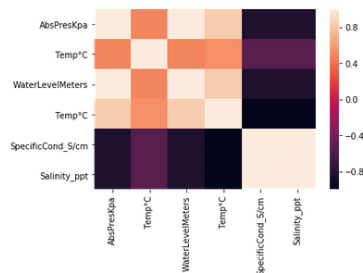
```
Out[14]:
```

	AbsPresKpa	Temp°C	WaterLevelMeters	Temp°C	SpecificCond_S/cm	Salinity_ppt
AbsPresKpa	1.000000	0.499190	0.999999	0.835213	-0.834792	-0.834750
Temp°C	0.499190	1.000000	0.499218	0.541818	-0.541607	-0.541576
WaterLevelMeters	0.999999	0.499218	1.000000	0.835290	-0.834868	-0.834827
Temp°C	0.835213	0.541818	0.835290	1.000000	-0.999999	-0.999999
SpecificCond_S/cm	-0.834792	-0.541607	-0.834868	-0.999999	1.000000	1.000000
Salinity_ppt	-0.834750	-0.541576	-0.834827	-0.999999	1.000000	1.000000

entre las variables, que durante la actividad nos interesamos por la temperatura y la salinidad respecto al nivel del mar.

```
In [15]: sns.heatmap( result,
                    xticklabels=result.columns,
                    yticklabels=result.columns)
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x17482ef34a8>
```



### 3 Conclusión

Gracias a Plotly nuestro análisis es mas sencillo por su propiedad de ser interactiva y mucho mas estética que otros graficadores en Python. Respecto a lo que lleva al análisis de datos en esta actividad, podemos encontrar dos relaciones muy importantes en el mapa de calor:

Correlación entre nivel del mar y Temperatura = 0.835290.

Correlación entre nivel del mar y Salinidad = -0.834827.

Y curiosamente la correlación entre Temperatura y Salinidad es casi del -1.00.

De estos datos se pueden concluir varias cosas:

La temperatura aumenta cuando sube el nivel del mar, porque entra agua de afuera del manglar que esta caliente por la radiación del sol. \*La salinidad disminuye al subir el nivel del mar porque