

UNIVERSIDAD DE SONORA

FÍSICA COMPUTACIONAL I

Evaluación 1

Autor:

Manuel I. GÓMEZ G.

Profesor:

Carlos LIZÁRRAGA
CELAYA

9 de marzo del 2018



**"El saber de mis hijos
hará mi grandeza"**

1 Introducción

El presente reporte evidencia todos los pasos que se siguieron a fin de concretar la primer evaluación semestral. A lo largo del presente documento se mostraran los resultados al manejar los datos (gráficas) y se contestarán ciertas preguntas con el fin de interpretar dichos resultados.

2 Descripción de la estructura de los datos

Los archivos que contenían todos los datos utilizados fueron *sargento_201117.csv* y *sargento_salinidad_201117.csv*. Ambos con diversas columnas que registraban mediciones del nivel del mar, temperatura del agua, presión, salinidad, conductancia y la fecha con hora de registro para cada uno de los datos.

En un principio existía una diferencia en la cantidad de datos entre ambos archivos, pero fue solucionada al momento de leer los archivos, comenzando la lectura desde otro punto y posteriormente eliminamos la columna que nos es innecesaria en ambos archivos.

```
#Leemos cada archivo
df0 = pd.read_csv("sargento_201117.csv", skiprows=1, sep=",", header=None, names=["#", "Date Time", "Pres (kPa)", "Temp (°C)", "Water Level (m)"])
df1 = pd.read_csv("sargento_salinidad_201117.csv", skiprows=1, sep=",", header=None, names=["#", "Date Time", "Cond High Rng (?S/cm)", "Temp (°C)", "Conductance (?S/cm)", "Salinity (ppt)"])
```

```
#Removemos la columna de valores
df0 = df0.drop(["#"], 1)
df0.head()
```

Ahora verificamos los datos capturados en el archivo con la función mostrada en la imagen anterior `df0.head()` para *sargento_201117.csv* y `df1.head()` para *sargento_salinidad_201117.csv*.

1) Datos de *sargento_201117.csv*

	Date Time	Pres (kPa)	Temp (°C)	Water Level (m)
0	10/26/2017 13:00:00	105.612	24.448	-0.150
1	10/26/2017 13:15:00	105.513	24.351	-0.160
2	10/26/2017 13:30:00	105.433	24.351	-0.168
3	10/26/2017 13:45:00	105.385	24.351	-0.173
4	10/26/2017 14:00:00	105.321	24.351	-0.179

2) Datos de sargento_salinidad_201117.csv

	Date Time	Cond High Rng (?S/cm)	Temp (°C)	Conductance (?S/cm)	Salinity (ppt)
0	10/26/2017 13:00:00	54525.5	24.91	54622.1	36.1588
1	10/26/2017 13:15:00	54525.5	24.82	54719.0	36.2311
2	10/26/2017 13:30:00	54525.5	24.76	54783.8	36.2794
3	10/26/2017 13:45:00	54525.5	24.75	54794.6	36.2875
4	10/26/2017 14:00:00	54525.5	24.73	54816.2	36.3036

3 Gráficas de caja

Haciendo uso la biblioteca **Seaborn** se nos pide que generemos tres distintas gráficas de caja. Antes de poder hacerlo debemos transformar la fecha (**Date Time**) que fue capturada como un objeto en una variable de tiempo (**Ndate** y **month**).

Se aplica el código tanto para **df0** como **df1**.

```
#Necesitamos que las fechas sean variables temporales y no objetos
df0['Ndate'] = pd.to_datetime(df0['Date Time'], format='%m/%d/%Y %H:%M:%S')
df0['month'] = df0['Ndate'].dt.month
df0.head()
```

	Date Time	Pres (kPa)	Temp (°C)	Water Level (m)	Ndate	month
0	10/26/2017 13:00:00	105.612	24.448	-0.150	2017-10-26 13:00:00	10
1	10/26/2017 13:15:00	105.513	24.351	-0.160	2017-10-26 13:15:00	10
2	10/26/2017 13:30:00	105.433	24.351	-0.168	2017-10-26 13:30:00	10
3	10/26/2017 13:45:00	105.385	24.351	-0.173	2017-10-26 13:45:00	10
4	10/26/2017 14:00:00	105.321	24.351	-0.179	2017-10-26 14:00:00	10

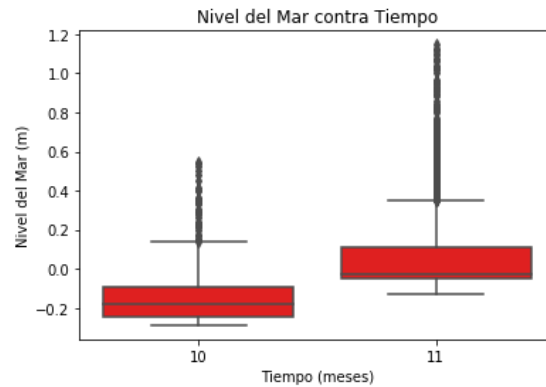
Ahora que tenemos una variable de tiempo aplicamos los siguientes códigos, según el caso:

I. Nivel del Mar (metros).

```
#Creamos la gráfica de NIVEL DEL MAR CONTRA TIEMPO
NMvsT = sns.boxplot(x="month", y="Water Level (m)", data=df0, color="red")

plt.title("Nivel del Mar contra Tiempo")
plt.ylabel("Nivel del Mar (m)")
plt.xlabel("Tiempo (meses)")

plt.show()
```

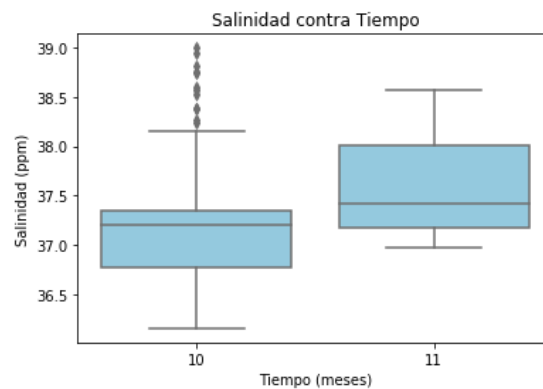


II. Salinidad (Partes por mil - ppm).

```
#Creamos la gráfica de SALINIDAD CONTRA TIEMPO
SvsT = sns.boxplot(x="month", y="Salinity (ppt)", data=df1, color="skyblue")

plt.title("Salinidad contra Tiempo")
plt.ylabel("Salinidad (ppm)")
plt.xlabel("Tiempo (meses)")

plt.show()
```

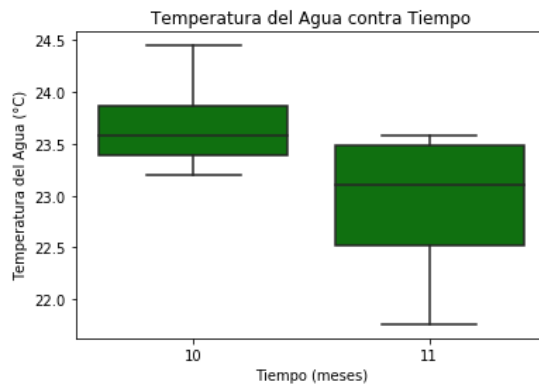


III. Temperatura del Agua (°C).

```
#Creamos la gráfica de TEMPERATURA DEL AGUA CONTRA TIEMPO
SvsT = sns.boxplot(x="month", y="Temp (°C)", data=df0, color="green")

plt.title("Temperatura del Agua contra Tiempo")
plt.ylabel("Temperatura del Agua (°C)")
plt.xlabel("Tiempo (meses)")

plt.show()
```



De igual forma es posible verificar o conocer específicamente los datos del gráfico si usamos la función **describe**, ya que al correr dicho código se nos genera una tabla que cuenta con los datos estadísticos generales para cada una de las columnas.

```
df0.describe()
```

	Pres (kPa)	Temp (°C)	Water Level (m)	month
count	2394.000000	2394.000000	2394.000000	2394.000000
mean	107.430007	23.120883	0.030863	10.781119
std	2.371844	0.563555	0.235974	0.413574
min	104.229000	21.760000	-0.288000	10.000000
25%	106.407000	22.525000	-0.071000	11.000000
50%	106.764000	23.388000	-0.035000	11.000000
75%	107.305000	23.484000	0.018750	11.000000
max	118.641000	24.448000	1.146000	11.000000

```
df1.describe()
```

	Cond High Rng (?S/cm)	Temp (°C)	Conductance (?S/cm)	Salinity (ppt)	month
count	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000
mean	54524.972807	23.316646	56386.831662	37.479737	10.781119
std	11.876669	0.547033	619.501987	0.464974	0.413574
min	54105.700000	21.490000	54622.100000	36.158800	10.000000
25%	54525.500000	22.730000	55949.700000	37.151400	11.000000
50%	54525.500000	23.490000	56185.600000	37.328300	11.000000
75%	54525.500000	23.700000	57053.700000	37.980300	11.000000
max	54525.500000	24.910000	58398.700000	38.994200	11.000000

4 Regresión lineal: correlación entre variables

Para llevar a cabo la regresión lineal y verificar si existe dicha correlación es necesario tener todos los datos en un sólo DataFrame, por lo que debemos de unificar **df0** y **df1** para crear **df2**.

Primero, eliminemos las columnas *Date Time*, *Temp (°C)*, *Ndate* y *month* de **df1** debido a que son columnas ya existentes en **df0**.

```
#Lo siguiente es unir df0 y df1 en un sólo DataFrame para poder generar las siguientes gráficas
#Primero hemos de remover aquellos datos repetidos en df1
df1=df1.drop(["Date Time","Temp (°C)","Ndate","month"],1)
df1.head()
```

	Cond High Rng (?S/cm)	Conductance (?S/cm)	Salinity (ppt)
0	54525.5	54622.1	36.1588
1	54525.5	54719.0	36.2311
2	54525.5	54783.8	36.2794
3	54525.5	54794.6	36.2875
4	54525.5	54816.2	36.3036

Ahora que las columnas repetidas han sido eliminadas es más sencillo unir los DataFrames. El código utilizado es el siguiente:

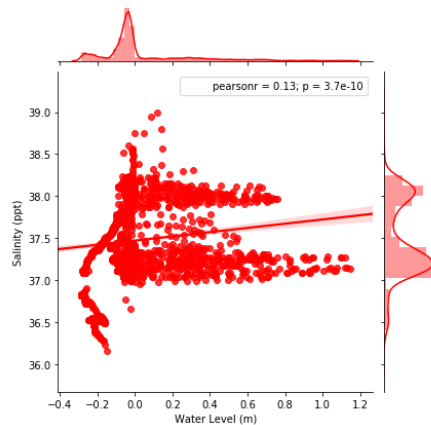
```
#Como ya removimos los datos repetidos podemos hacer la unión en un nuevo DataFrame
df2=pd.concat([df0, df1], axis=1, join_axes=[df1.index])
df2.head()
```

	Date Time	Pres (kPa)	Temp (°C)	Water Level (m)	Ndate	month	Cond High Rng (?S/cm)	Conductance (?S/cm)	Salinity (ppt)
0	10/26/2017 13:00:00	105.612	24.448	-0.150	2017-10-26 13:00:00	10	54525.5	54622.1	36.1588
1	10/26/2017 13:15:00	105.513	24.351	-0.160	2017-10-26 13:15:00	10	54525.5	54719.0	36.2311
2	10/26/2017 13:30:00	105.433	24.351	-0.168	2017-10-26 13:30:00	10	54525.5	54783.8	36.2794
3	10/26/2017 13:45:00	105.385	24.351	-0.173	2017-10-26 13:45:00	10	54525.5	54794.6	36.2875
4	10/26/2017 14:00:00	105.321	24.351	-0.179	2017-10-26 14:00:00	10	54525.5	54816.2	36.3036

Tras haber hecho eso, podemos proseguir a realizar la regresión lineal usando **Seaborn**. Para generarlas usaremos los siguientes código:

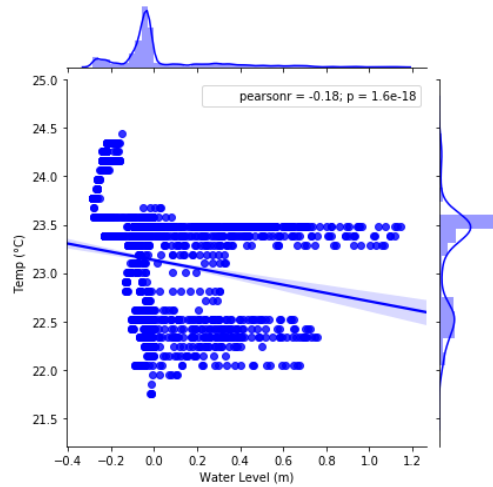
I. Nivel del Mar - Salinidad.

```
#Correlación entre NIVEL DEL MAR-SALINIDAD
#sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot(x="Water Level (m)", y="Salinity (ppt)", data=df2, kind="reg", color="r", size=7)
plt.show()
```



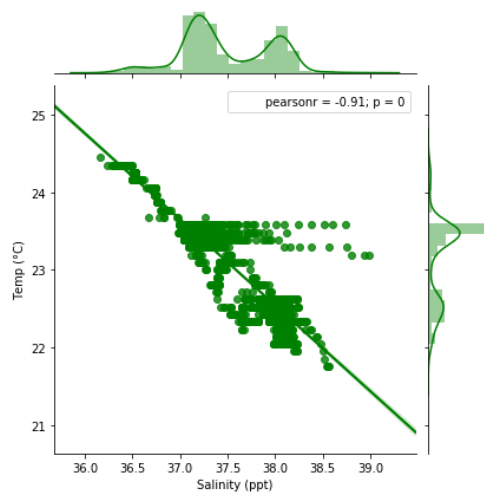
II. Nivel de Mar - Temperatura del Agua

```
#Correlación entre NIVEL DEL MAR-TEMPERATURA DEL AGUA
#sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot(x="Water Level (m)", y="Temp (°C)", data=df2, kind="reg", color="b", size=7)
plt.show()
```



III. Salinidad - Temperatura del Agua.

```
#Correlación entre SALINIDAD-TEMPERATURA DEL AGUA
#sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot(x="Salinity (ppt)", y="Temp (°C)", data=df2, kind="reg", color="g", size=7)
plt.show()
```



Al observar las gráficas vemos los distintos comportamientos que tiene una variable con respecto a la otra y en base a ello determinamos que no existe una correlación entre Nivel del Mar-Salinidad y Nivel del Mar-Temperatura del Agua, sin embargo, la última de ellas muestra que efectivamente entre la salinidad y la temperatura del agua se aprecia dicha correlación.

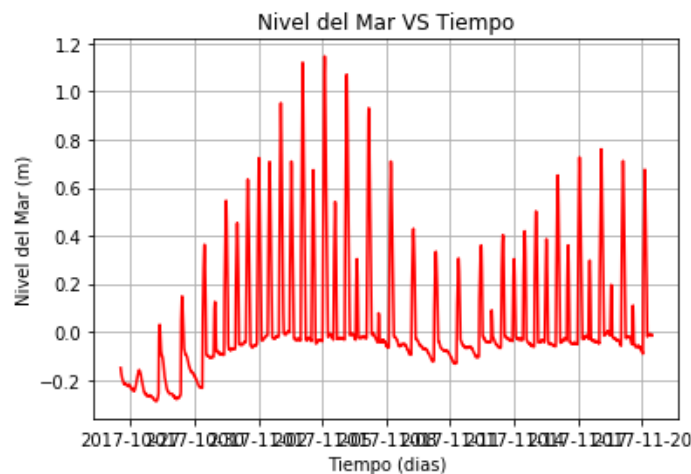
5 Gráficas contra el tiempo

En este punto se nos pide generar gráficas para cada una de las variables con las que hemos estado trabajando (Nivel del Mar, Temperatura del Agua y Salinidad) en función del tiempo.

Esta sección de la actividad es muy parecida pues sólo basta con cambiar los nombres de las variables para generar las distintas gráficas.

I. Nivel del Mar en función del Tiempo

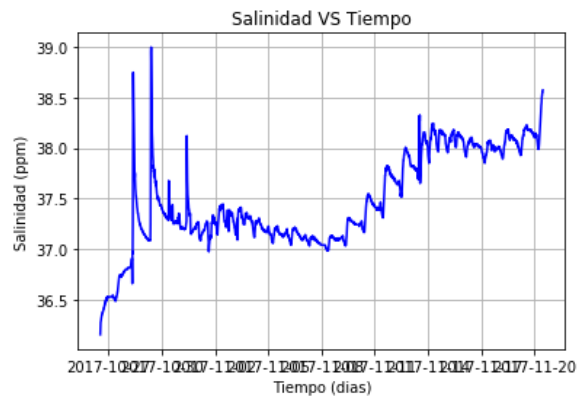
```
#El siguiente paso indica crear tres gráficas distintas:  
#Nivel del Mar VS Tiempo  
WaterLevel=df2["Water Level (m)"]  
Time=df2["Ndate"]  
  
plt.plot_date(x=Time, y=WaterLevel, fmt="r-")  
plt.title("Nivel del Mar VS Tiempo")  
plt.ylabel("Nivel del Mar (m)")  
plt.xlabel("Tiempo (dias)")  
plt.grid(True)  
plt.show()
```



II. Salinidad en función del Tiempo

```
#Salinidad VS Tiempo
Salinity=df2["Salinity (ppt)"]

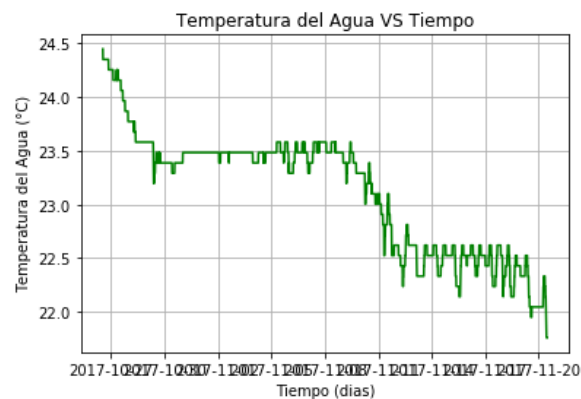
plt.plot_date(x=Time, y=Salinity, fmt="b-")
plt.title("Salinidad VS Tiempo")
plt.ylabel("Salinidad (ppm)")
plt.xlabel("Tiempo (dias)")
plt.grid(True)
plt.show()
```



III. Temperatura del Agua en función del Tiempo

```
#Temperatura del Agua VS Tiempo
TemperaturaAgua=df2["Temp (°C)"]

plt.plot_date(x=Time, y=TemperaturaAgua, fmt="g-")
plt.title("Temperatura del Agua VS Tiempo")
plt.ylabel("Temperatura del Agua (°C)")
plt.xlabel("Tiempo (dias)")
plt.grid(True)
plt.show()
```



6 Gráficas de doble eje

Este tipo de gráficas ahora implican sobreponer los datos de dos de nuestras variables permitiendonos observarlas en el eje y a lo largo del mismo intervalo de tiempo.

La diferencia aquí sería el establecer que es una gráfica de doble eje. En el siguiente código se muestra cómo se hizo.

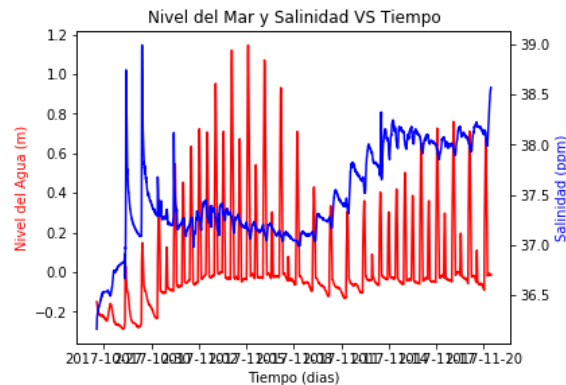
I. Nivel del Mar y Salinidad

```
#Ahora crearemos gráficas con un eje vertical doble...
#Primero, Nivel del Mar y Salinidad VS Tiempo
NM_S, ax1 = plt.subplots()

ax1.plot(Time, WaterLevel, "r-")
ax1.set_xlabel("Tiempo (dias)")
ax1.set_ylabel("Nivel del Agua (m)", color="r")

ax2=ax1.twinx()
ax2.plot(Time, Salinity, "b-")
ax2.set_ylabel("Salinidad (ppm)", color="b")

NM_S.tight_layout()
plt.title("Nivel del Mar y Salinidad VS Tiempo")
plt.show()
```



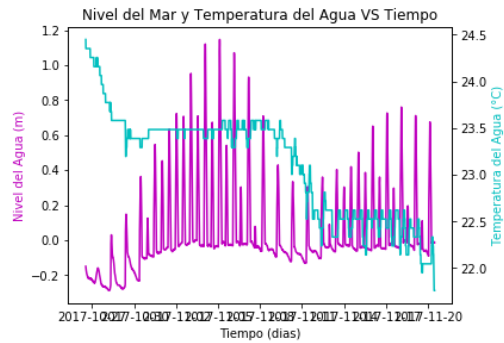
II. Nivel del Mar y Temperatura

```
#Segundo, Nivel del Mar y Temperatura del Agua VS Tiempo
NM_S, ax1 = plt.subplots()

ax1.plot(Time, WaterLevel, "m-")
ax1.set_xlabel("Tiempo (dias)")
ax1.set_ylabel("Nivel del Agua (m)", color="m")

ax2=ax1.twinx()
ax2.plot(Time, TemperaturaAgua, "c-")
ax2.set_ylabel("Temperatura del Agua (°C)", color="c")

NM_S.tight layout()
plt.title("Nivel del Mar y Temperatura del Agua VS Tiempo")
plt.show()
```



7 Estableciendo límites

Por último, debemos de crear nuevamente las gráficas de eje doble con la pequeña modificación de que ahora delimitaremos el eje del tiempo a solamente 5 días de la muestra total, de este modo podremos ver más a detalle el comportamiento para posteriormente dar nuestra conclusión acerca de las gráficas.

El código empleado para esta actividad fue el siguiente:

I. Nivel del Mar y Salinidad

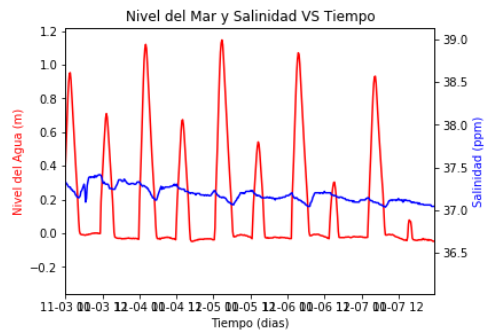
```
#Nuestro último requisito es analizar nuevamente las gráficas anteriores en un intervalo de 5 días
#Primero, Nivel del Mar y Salinidad VS Tiempo
NM_S, ax1 = plt.subplots()

ax1.plot(Time, WaterLevel, "r-")
ax1.set_xlabel("Tiempo (dias)")
ax1.set_ylabel("Nivel del Agua (m)", color="r")

ax2=ax1.twinx()
ax2.plot(Time, Salinity, "b-")
ax2.set_ylabel("Salinidad (ppm)", color="b")

plt.xlim(["2017-11-03 00:00:00", "2017-11-07 23:45:00"])

NM_S.tight_layout()
plt.title("Nivel del Mar y Salinidad VS Tiempo")
plt.show()
```



II. Nivel del Mar y Temperatura

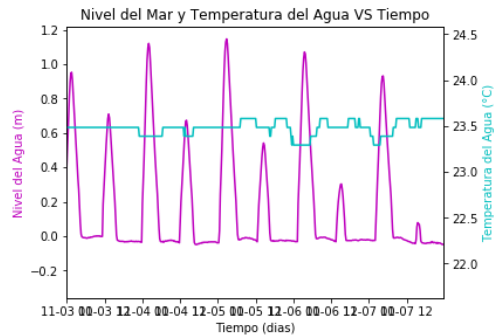
```
#Segundo, Nivel del Mar y Temperatura del Agua VS Tiempo
NM_S, ax1 = plt.subplots()

ax1.plot(Time, WaterLevel, "m-")
ax1.set_xlabel("Tiempo (dias)")
ax1.set_ylabel("Nivel del Agua (m)", color="m")

ax2=ax1.twinx()
ax2.plot(Time, TemperaturaAgua, "c-")
ax2.set_ylabel("Temperatura del Agua (°C)", color="c")

plt.xlim(["2017-11-03 00:00:00", "2017-11-07 23:45:00"])

NM_S.tight_layout()
plt.title("Nivel del Mar y Temperatura del Agua VS Tiempo")
plt.show()
```



Ahora que podemos apreciar más detalladamente las gráficas hemos de notar la relación existente entre las variables.

A partir del gráfico del Nivel del Mar y Salinidad en función del Tiempo podríamos decir que en el momento en que aumenta el Nivel del Mar, lo cual implica que aumenta la marea, la salinidad se reduce, esto podría ser por la forma en que se ve distribuida en presencia de agua dulce.

La gráfica del Nivel del Mar y Temperatura del Agua en función del Tiempo muestra que a la llegada de agua dulce con el aumento de la marea la temperatura desciende unas cuantas décimas de grado centígrado, en cuanto no llegue nuevamente agua está aumenta de igual forma una pequeña fracción de grado.