Evaluación 1

Brenda Leyva Amaya

8 de Marzo, 2018

1 Archivos de datos.

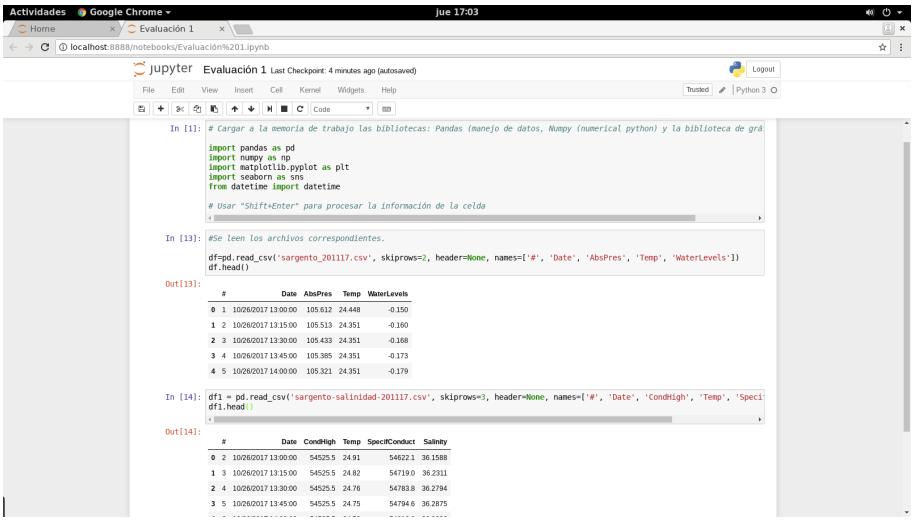
Se han decargado dos archivos que guardan datos de las distintas tomas del equipo en la estación, uno de ellos es referente a la salinidad y el otro contiene datos más generales de la zona.

Los archivos tienen la misma estructura de columnas y los datos corresponden a tomas cada 15 minutos con su fecha correspondiente. Estos archivos se han proporcionado de manera que pueden ser leídos por jupyter. No ha sido necesario llevar a cabo una limpieza de los mismos.

De manera que se procede directamente a leerlos y crear los correpondientes data frames.

2 Uso de Jupyter Notebook.

Utilizamos Jupyter Notebook para llevar a cabo análisis de los datos. A continuación se describen y muestras las actividades que se llevaron a cabo. Primero que nada se ha hecho la lectura de los archivos en jupyter y después se comenzó el análisis de datos.



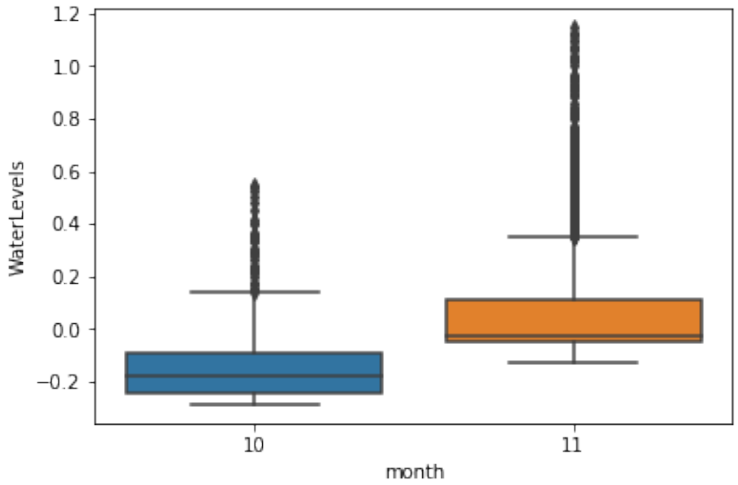
2.1 Biblioteca Seaborn.

El primer paso será abordar la herramienta Seaborn para crear gráficos y ejercicios correspondientes.

2.1.1 Gráficas de caja.

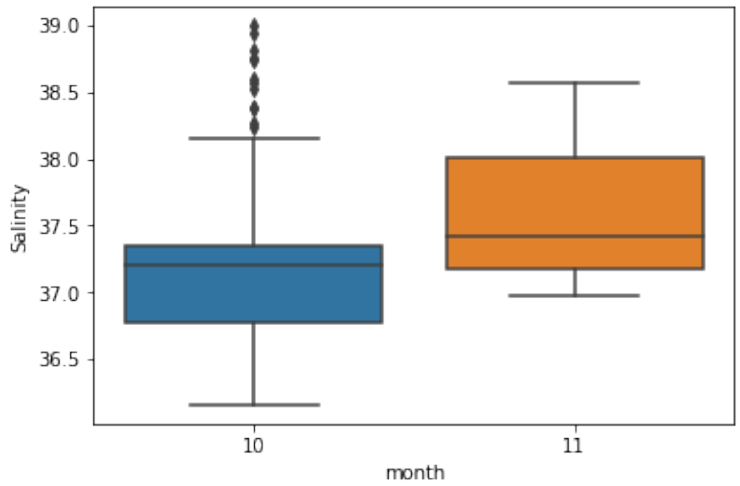
a) Nivel de mar (metros)

```
box1 = sns.boxplot(x="month", y="WaterLevels", data=df)
plt.show()
```



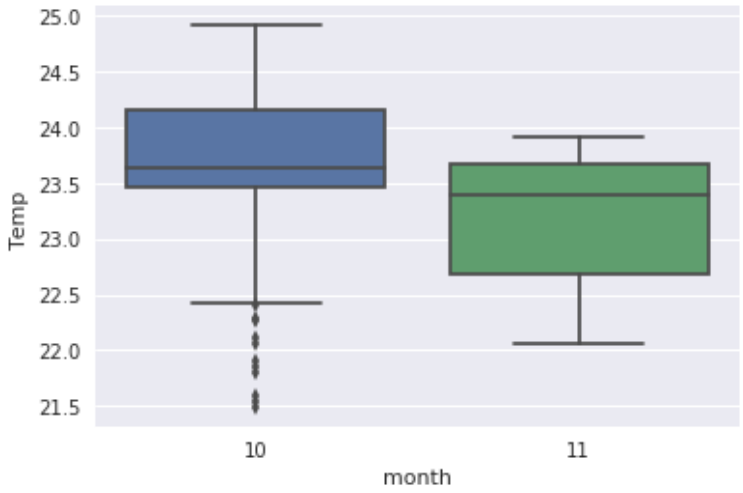
b) Salinidad (Partes por mil - ppt)

```
box2 = sns.boxplot(x="month", y="Salinity", data=df1)
plt.show()
```



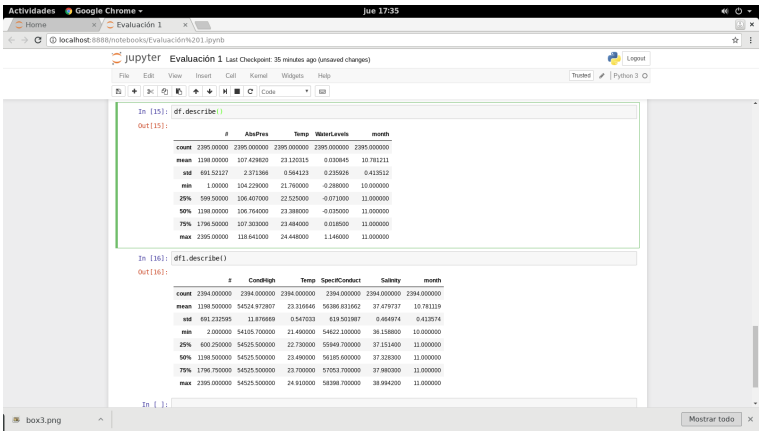
c) Temperatura de Agua (°C).

```
box3 = sns.boxplot(x="month", y="Temp", data=df1)
plt.show()
```



2.1.2 Función "Describe".

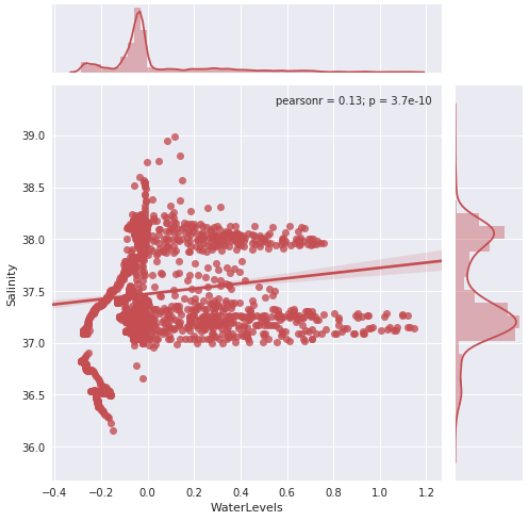
A continuación se presenta el análisis estadístico con el uso de la función describe.



2.1.3 Correlaciones.

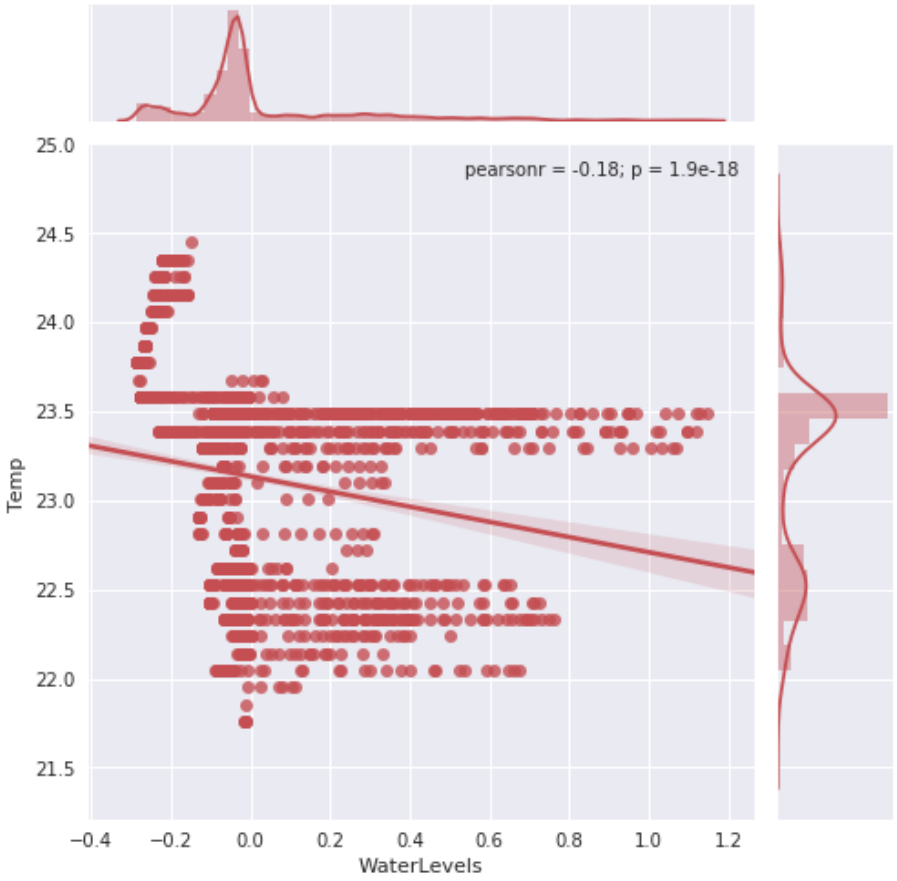
Nivel de mar-Salinidad.

```
sns.set(style="darkgrid", color_codes=True)

ws = sns.jointplot("WaterLevels", "Salinity", data=pd.concat([df,df1],axis=1,join_axes=[df1.
plt.show(ws)
```



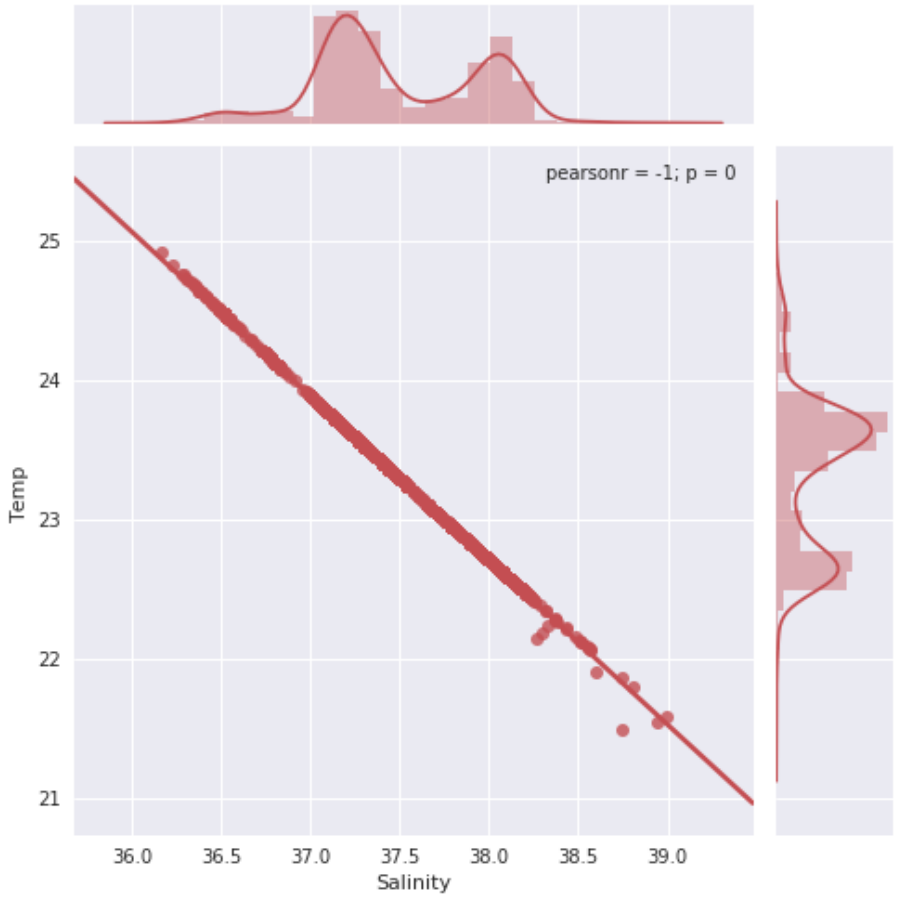
Nivel de mar-Temperatura del agua.

```
sns.set(style="darkgrid", color_codes=True)

wt = sns.jointplot("WaterLevels", "Temp", data=df, kind="reg",color="r", size=7)
plt.show(wt)
```



Salinidad-Temperatura del agua.

```
sns.set(style="darkgrid", color_codes=True)

st = sns.jointplot("Salinity", "Temp", data=df1, kind="reg",color="r", size=7)
plt.show(st)
```



2.2 Matplotlib.

A continuación utilizaremos la herramienta Matplotlib para realizar las siguientes actividades.

2.2.1 Variables VS tiempo.

Nivel del mar

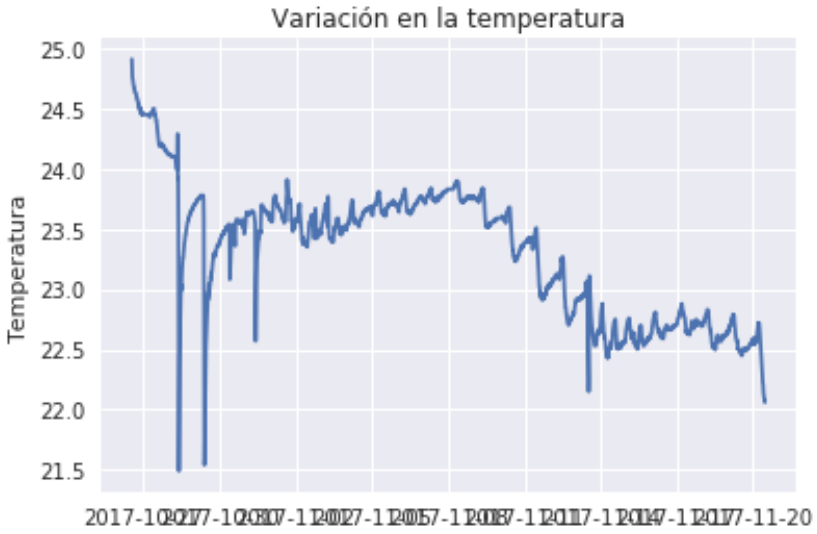
```
plt.plot_date(x=df.Ndate, y=df.WaterLevels, fmt="b-")
plt.title("Variación de los niveles de agua")
plt.ylabel("Niveles de agua")
plt.grid(True)
plt.show()
```

Salinidad

```
plt.plot_date(x=df1.Ndate, y=df1.Salinity, fmt="b-")
plt.title("Variación de la salinidad")
plt.ylabel("Salinidad")
plt.grid(True)
plt.show()
```

Temperatura del Agua

```
plt.plot_date(x=df1.Ndate, y=df1.Temp, fmt="b-")
plt.title("Variación en la temperatura")
plt.ylabel("Temperatura")
plt.grid(True)
plt.show()
```

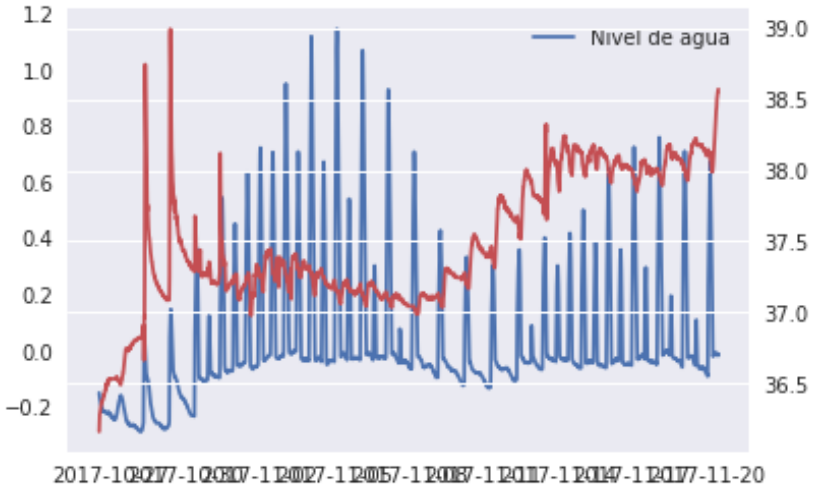


2.2.2 Gráficas superpuestas.

a) Nivel de mar y Salinidad

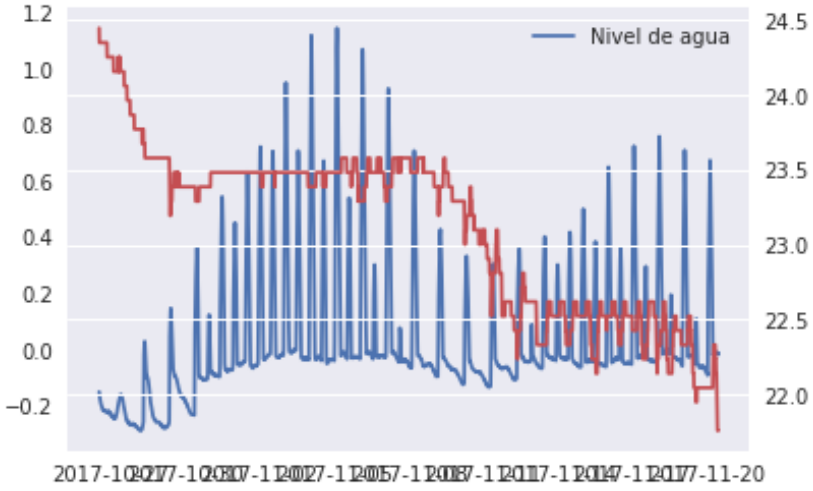
```
from matplotlib import rc
rc('mathtext', default='regular')

df2 = pd.concat([df.WaterLevels, df1.Salinity],axis=1)

fig = plt.figure()
ax = fig.add_subplot(111)

Fecha=df.Ndate
WL=df.WaterLevels
SAL=df1.Salinity

ax.plot(Fecha, WL, '-b', label = 'Nivel de agua')

ax2 = ax.twinx()
```

```
ax2.plot(Fecha, SAL, '-r', label = 'Salinidad')

ax.legend(loc=0)
ax.grid()

plt.show()
```



b) Nivel de mar y Temperatura.

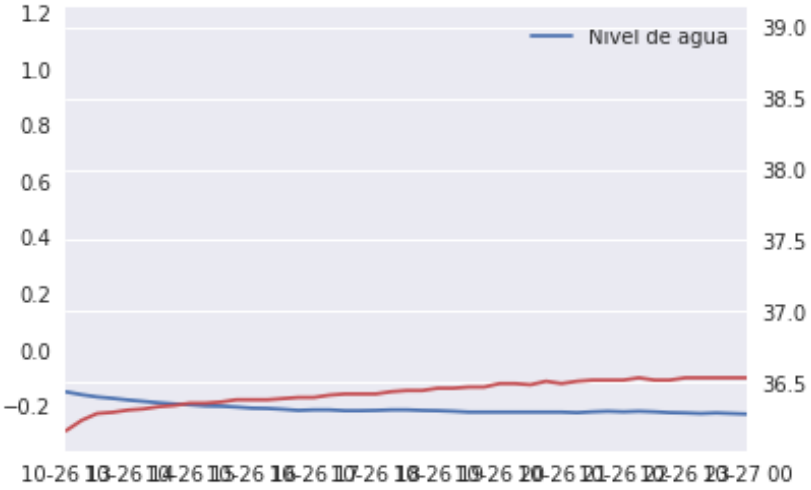
```
from matplotlib import rc
rc('mathtext', default='regular')

fig = plt.figure()
ax = fig.add_subplot(111)

Fecha=df.Ndate
WL=df.WaterLevels
Temp=df.Temp

ax.plot(Fecha, WL, '-b', label = 'Nivel de agua')

ax2 = ax.twinx()
ax2.plot(Fecha, Temp, '-r', label = 'Temperatura')

ax.legend(loc=0)
ax.grid()

plt.show()
```



Para 5 días:
a) Nivel de mar y Salinidad

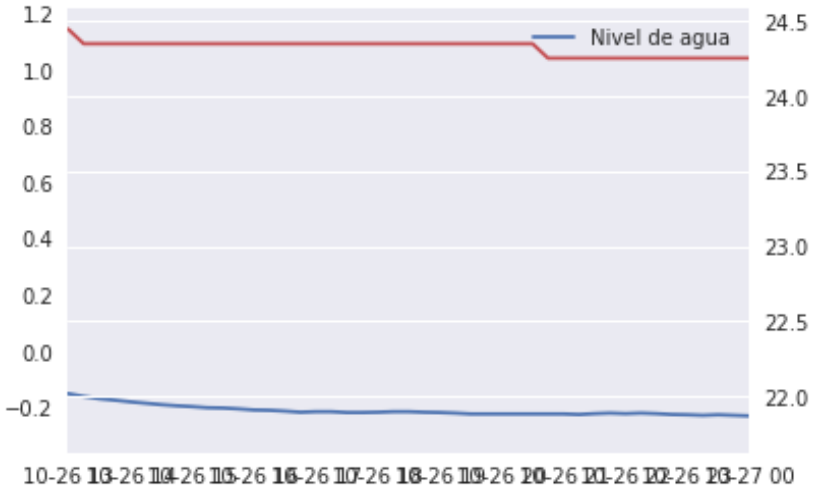
```
from matplotlib import rc
rc('mathtext', default='regular')

df2 = pd.concat([df.WaterLevels, df1.Salinity],axis=1)

fig = plt.figure()
ax = fig.add_subplot(111)

Fecha=df.Ndate
WL=df.WaterLevels
SAL=df1.Salinity

ax.plot(Fecha, WL, '-b', label = 'Nivel de agua')

ax2 = ax.twinx()
ax2.plot(Fecha, SAL, '-r', label = 'Salinidad')

ax2.set_xlim("10/26/2017 13:00:00","10/27/2017 00:00:00")
ax.set_xlim("10/26/2017 13:00:00","10/27/2017 00:00:00")


ax.legend(loc=0)
ax.grid()
```

```
plt.show()
```



b) Nivel de mar y Temperatura.

```
from matplotlib import rc
rc('mathtext', default='regular')

fig = plt.figure()
ax = fig.add_subplot(111)

Fecha=df.Ndate
WL=df.WaterLevels
Temp=df.Temp

ax.plot(Fecha, WL, '-b', label = 'Nivel de agua')

ax2 = ax.twinx()
ax2.plot(Fecha, Temp, '-r', label = 'Temperatura')

ax2.set_xlim("10/26/2017 13:00:00","10/27/2017 00:00:00")
ax.set_xlim("10/26/2017 13:00:00","10/27/2017 00:00:00")

ax.legend(loc=0)
ax.grid()

plt.show()
```



3 Conclusiones.

La actividad se llevó a cabo satisfactoriamente, se encontraron datos interesantes sobre todo en la correlación de la salinidad del agua con su temperatura pues esta se observó en su comportamiento gráfico como una relación casi perfectamente lineal, con muy pocos datos fuera de la interpolación. Se espera que las actividades realizadas se hayan llevado a cabo de manera correcta y que cumplan las metas fijadas para la evaluación.