

# Actividad 2

Juan Pedro Barajas

A 07 de Febrero del 2018

## 1 Introducción a Python

Como parte de la segunda actividad de la materia de física computacional 1, tuvimos como objetivo principal la aventura de conocer Python por primera vez.

Para comenzar con la actividad de esta semana comenzamos involucrandonos en el ambiente de jupyter notebook la cual es el lugar donde nos desembolveremos en el código python, aquí nos encontramos con la sorpresa de que programar en este lenguaje de programación es más fácil ya que la interfaz visual es mucho más amigable que los idiomas que he utilizado anteriormente, como lo son fortran y C++, además con la ventaja de que esta se puede traducir a lenguaje de LaTeX como a continuación mostraré.

En primera parte analizamos el comportamiento climático de una ciudad, guiándonos del ejemplo dado por el profesor escogimos una ciudad de la república mexicana donde quisiéramos realizar nuestro análisis.

### 1.1 Primera parte de Python

En este análisis de datos vemos cada parte del código donde los comentarios (iniciados con #) describen lo que se hace en cada parte del código, donde si, habrá algunas partes con error pero es de las primeras ocasiones en las que utilicé este lenguaje y repito, el código utilizado aquí para la configuración de color lo descargué directamente de jupyter.

```
In [1]: #ejemplo
        # Cargar a la memoria de trabajo las bibliotecas: Pandas (manejo de datos,
        # Numpy (numerical python) y la biblioteca de gráficas Matplotlib
        # Se asignan nombres cortos.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

        # Usar "Shift+Enter" para procesar la información de la celda
        #

In [2]: # Descarga los datos de una estación del Servicio Meteorológico Nacional
        # http://smn1.conagua.gob.mx/emas/
```

```
# Lee un archivo de texto con la función Pandas "read_csv", con elementos separados por
# un espacio, brincándose 4 renglones del inicio (encabezados)
df0 = pd.read_csv('Acaponeta.TXT', skiprows=4, sep='\s+')
# "Shift + Enter"
```

```
In [3]: # Descarga los datos de una estación del Servicio Meteorológico Nacional
# http://smn1.conagua.gob.mx/emas/
# Lee un archivo de texto con la función Pandas "read_csv", con elementos separados por
# un espacio, brincándose 4 renglones del inicio (encabezados)
df0 = pd.read_csv('Acaponeta.TXT', skiprows=4, sep='\s+')
# "Shift + Enter"
```

```
In [4]: # Lee los primeros 5 renglones del archivo
df0.head()
# "Shift+Enter"
```

```
Out[4]:
```

|   | DD/MM/AAAA | HH:MM | DIRS  | DIRR  | VELS | VELR | TEMP | HR   | PB     | PREC | \ |
|---|------------|-------|-------|-------|------|------|------|------|--------|------|---|
| 0 | 25/01/2018 | 22:00 | 183.0 | 339.0 | 0.95 | 25.9 | 23.5 | 65.0 | 1011.8 | 0.0  |   |
| 1 | 25/01/2018 | 23:00 | 148.0 | 349.0 | 5.39 | 18.7 | 23.9 | 64.0 | 1011.9 | 0.0  |   |
| 2 | 26/01/2018 | 00:00 | 86.0  | 9.0   | 6.27 | 14.8 | 23.9 | 60.0 | 1012.1 | 0.0  |   |
| 3 | 26/01/2018 | 01:00 | 54.0  | 7.0   | 6.24 | 16.6 | 23.3 | 60.0 | 1012.5 | 0.0  |   |
| 4 | 26/01/2018 | 02:00 | 290.0 | 342.0 | 3.79 | 14.8 | 22.7 | 64.0 | 1012.8 | 0.0  |   |

```
RADSOL
0    107.8
1     92.8
2     16.7
3     -1.0
4     -0.8
```

```
In [5]: # Dar estructura de datos (DataFrame)
df = pd.DataFrame(df0)
```

```
In [6]: # Ver los tipos de datos que Pandas ha reconocido al leer
df.dtypes
```

```
Out[6]: DD/MM/AAAA      object
HH:MM      object
DIRS      float64
DIRR      float64
VELS      float64
VELR      float64
TEMP      float64
HR      float64
PB      float64
PREC      float64
RADSOL      float64
dtype: object
```

```
In [7]: # Combinar las columnas "DD/MM/AAAA" con "HH:MM" y convertirla a variable de tiempo.
# Se crea una nueva columna "Fecha" al final con formato de tiempo.
# Eliminamos las dos primeras columnas que ya no necesitaremos
df['FECHA'] = pd.to_datetime(df.apply(lambda x: x['DD/MM/AAAA'] + ' ' + x['HH:MM'], axis=1))
df = df.drop(['DD/MM/AAAA', 'HH:MM'], 1)
```

```
In [8]: df.head(14)
```

```
Out [8]:
```

|    | DIRS  | DIRR  | VELS | VELR | TEMP | HR   | PB     | PREC | RADSOL | \ |
|----|-------|-------|------|------|------|------|--------|------|--------|---|
| 0  | 183.0 | 339.0 | 0.95 | 25.9 | 23.5 | 65.0 | 1011.8 | 0.0  | 107.8  |   |
| 1  | 148.0 | 349.0 | 5.39 | 18.7 | 23.9 | 64.0 | 1011.9 | 0.0  | 92.8   |   |
| 2  | 86.0  | 9.0   | 6.27 | 14.8 | 23.9 | 60.0 | 1012.1 | 0.0  | 16.7   |   |
| 3  | 54.0  | 7.0   | 6.24 | 16.6 | 23.3 | 60.0 | 1012.5 | 0.0  | -1.0   |   |
| 4  | 290.0 | 342.0 | 3.79 | 14.8 | 22.7 | 64.0 | 1012.8 | 0.0  | -0.8   |   |
| 5  | 106.0 | 29.0  | 5.96 | 14.4 | 22.3 | 66.0 | 1013.0 | 0.0  | -1.0   |   |
| 6  | 63.0  | 7.0   | 9.14 | 16.2 | 22.4 | 64.0 | 1013.1 | 0.0  | -1.0   |   |
| 7  | 107.0 | 23.0  | 6.65 | 18.4 | 22.2 | 63.0 | 1013.2 | 0.0  | -1.0   |   |
| 8  | 134.0 | 15.0  | 5.68 | 16.9 | 22.0 | 63.0 | 1013.0 | 0.0  | -1.0   |   |
| 9  | 176.0 | 356.0 | 8.06 | 18.0 | 21.3 | 65.0 | 1012.7 | 0.0  | -1.0   |   |
| 10 | 63.0  | 15.0  | 2.88 | 17.3 | 20.6 | 68.0 | 1012.3 | 0.0  | -1.0   |   |
| 11 | 41.0  | 8.0   | 1.76 | 15.5 | 19.7 | 70.0 | 1011.5 | 0.0  | -1.0   |   |
| 12 | 249.0 | 30.0  | 1.78 | 16.6 | 19.6 | 69.0 | 1010.9 | 0.0  | -1.0   |   |
| 13 | 107.0 | 15.0  | 2.27 | 13.3 | 19.0 | 69.0 | 1010.8 | 0.0  | -1.0   |   |

|    | FECHA               |
|----|---------------------|
| 0  | 2018-01-25 22:00:00 |
| 1  | 2018-01-25 23:00:00 |
| 2  | 2018-01-26 00:00:00 |
| 3  | 2018-01-26 01:00:00 |
| 4  | 2018-01-26 02:00:00 |
| 5  | 2018-01-26 03:00:00 |
| 6  | 2018-01-26 04:00:00 |
| 7  | 2018-01-26 05:00:00 |
| 8  | 2018-01-26 06:00:00 |
| 9  | 2018-01-26 07:00:00 |
| 10 | 2018-01-26 08:00:00 |
| 11 | 2018-01-26 09:00:00 |
| 12 | 2018-01-26 10:00:00 |
| 13 | 2018-01-26 11:00:00 |

```
In [9]: df.dtypes
```

```
Out [9]: DIRS          float64
DIRR          float64
VELS          float64
VELR          float64
```

```

TEMP          float64
HR            float64
PB            float64
PREC          float64
RADSOL        float64
FECHA         datetime64[ns]
dtype: object

```

```

In [10]: # Realiza un análisis exploratorio de datos
df.describe()

```

```

Out[10]:

```

|       | DIRS        | DIRR       | VELS       | VELR       | TEMP \     |
|-------|-------------|------------|------------|------------|------------|
| count | 166.000000  | 166.000000 | 165.000000 | 165.000000 | 165.000000 |
| mean  | 160.038554  | 144.200602 | 5.285636   | 14.347273  | 22.446667  |
| std   | 107.303216  | 119.573799 | 3.509094   | 5.830790   | 5.855647   |
| min   | 29.000000   | 2.000000   | 0.410000   | 4.700000   | 11.900000  |
| 25%   | 77.750000   | 34.000000  | 2.760000   | 10.100000  | 18.500000  |
| 50%   | 142.500000  | 109.000000 | 4.520000   | 13.300000  | 20.900000  |
| 75%   | 233.750000  | 245.500000 | 7.030000   | 17.300000  | 27.000000  |
| max   | 1011.400000 | 424.300000 | 18.810000  | 33.100000  | 37.100000  |

|       | HR         | PB          | PREC  | RADSOL     |
|-------|------------|-------------|-------|------------|
| count | 165.000000 | 165.000000  | 165.0 | 165.000000 |
| mean  | 59.993939  | 1010.161212 | 0.0   | 162.000000 |
| std   | 19.709010  | 1.773710    | 0.0   | 245.541393 |
| min   | 21.000000  | 1006.100000 | 0.0   | -1.000000  |
| 25%   | 42.000000  | 1009.100000 | 0.0   | -1.000000  |
| 50%   | 64.000000  | 1010.100000 | 0.0   | -0.300000  |
| 75%   | 75.000000  | 1011.300000 | 0.0   | 278.000000 |
| max   | 96.000000  | 1014.300000 | 0.0   | 760.800000 |

```

In [11]: # Selecciona los renglones con Temperatura > 24°C y < 25°C
df_tmp = df[df.TEMP > 24]
df_select = df_tmp[df_tmp.TEMP < 25]
df_select

```

```

Out[11]:

```

|     | DIRS  | DIRR  | VELS | VELR | TEMP | HR   | PB     | PREC | RADSOL \ |
|-----|-------|-------|------|------|------|------|--------|------|----------|
| 27  | 266.0 | 293.0 | 2.11 | 13.7 | 24.2 | 56.0 | 1009.2 | 0.0  | -1.0     |
| 67  | 174.0 | 168.0 | 5.91 | 13.7 | 24.3 | 53.0 | 1010.4 | 0.0  | 637.7    |
| 113 | 94.0  | 14.0  | 4.53 | 12.2 | 24.1 | 44.0 | 1010.9 | 0.0  | 164.8    |
| 140 | 123.0 | 17.0  | 1.46 | 15.5 | 24.3 | 51.0 | 1012.2 | 0.0  | 141.5    |

|     | FECHA               |
|-----|---------------------|
| 27  | 2018-01-27 01:00:00 |
| 67  | 2018-01-28 17:00:00 |
| 113 | 2018-01-30 15:00:00 |
| 140 | 2018-01-31 18:00:00 |

```
In [12]: # Selecciona los renglones con Temperatura > 24°C y < 25°C
df_tmp = df[df.TEMP > 24]
df_select = df_tmp[df_tmp.TEMP < 25]
df_select
```

```
Out[12]:
```

|     | DIRS  | DIRR  | VELS | VELR | TEMP | HR   | PB     | PREC | RADSOL | \ |
|-----|-------|-------|------|------|------|------|--------|------|--------|---|
| 27  | 266.0 | 293.0 | 2.11 | 13.7 | 24.2 | 56.0 | 1009.2 | 0.0  | -1.0   |   |
| 67  | 174.0 | 168.0 | 5.91 | 13.7 | 24.3 | 53.0 | 1010.4 | 0.0  | 637.7  |   |
| 113 | 94.0  | 14.0  | 4.53 | 12.2 | 24.1 | 44.0 | 1010.9 | 0.0  | 164.8  |   |
| 140 | 123.0 | 17.0  | 1.46 | 15.5 | 24.3 | 51.0 | 1012.2 | 0.0  | 141.5  |   |

```
FECHA
27 2018-01-27 01:00:00
67 2018-01-28 17:00:00
113 2018-01-30 15:00:00
140 2018-01-31 18:00:00
```

```
In [13]: # Selecciona los renglones con Temperatura > 24°C y < 25°C
df_tmp = df[df.TEMP > 24]
df_select = df_tmp[df_tmp.TEMP < 25]
df_select
```

```
Out[13]:
```

|     | DIRS  | DIRR  | VELS | VELR | TEMP | HR   | PB     | PREC | RADSOL | \ |
|-----|-------|-------|------|------|------|------|--------|------|--------|---|
| 27  | 266.0 | 293.0 | 2.11 | 13.7 | 24.2 | 56.0 | 1009.2 | 0.0  | -1.0   |   |
| 67  | 174.0 | 168.0 | 5.91 | 13.7 | 24.3 | 53.0 | 1010.4 | 0.0  | 637.7  |   |
| 113 | 94.0  | 14.0  | 4.53 | 12.2 | 24.1 | 44.0 | 1010.9 | 0.0  | 164.8  |   |
| 140 | 123.0 | 17.0  | 1.46 | 15.5 | 24.3 | 51.0 | 1012.2 | 0.0  | 141.5  |   |

```
FECHA
27 2018-01-27 01:00:00
67 2018-01-28 17:00:00
113 2018-01-30 15:00:00
140 2018-01-31 18:00:00
```

```
In [14]: # Calcula el promedio de las columnas, excepto en la FECHA (que no tendría sentido)
df.mean()
```

```
Out[14]:
```

|        |             |
|--------|-------------|
| DIRS   | 160.038554  |
| DIRR   | 144.200602  |
| VELS   | 5.285636    |
| VELR   | 14.347273   |
| TEMP   | 22.446667   |
| HR     | 59.993939   |
| PB     | 1010.161212 |
| PREC   | 0.000000    |
| RADSOL | 162.000000  |

```
dtype: float64
```

```
In [15]: # Calcula el promedio de las Temperaturas
df.TEMP.mean()
```

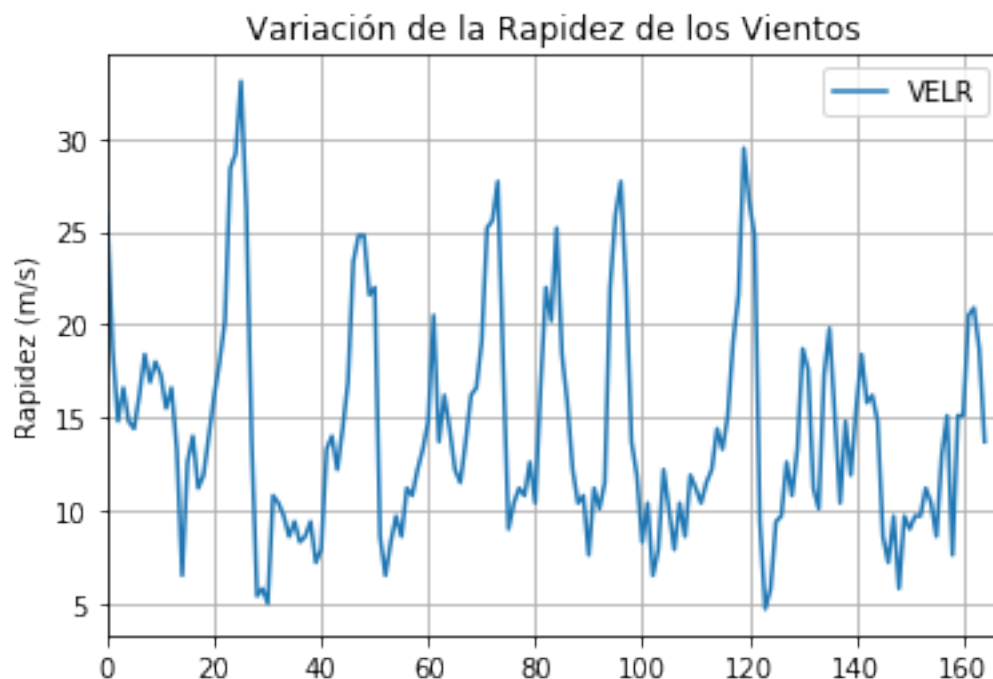
```
Out[15]: 22.446666666666666
```

## 1.2 Graficación

Como segunda parte del analisis, pasamos a realizar las graficas correspondientes a cada parte del comportamiento, como primeramente se pueden ver la grafica de la rapidez de los vientos y de las rafagas de viento.

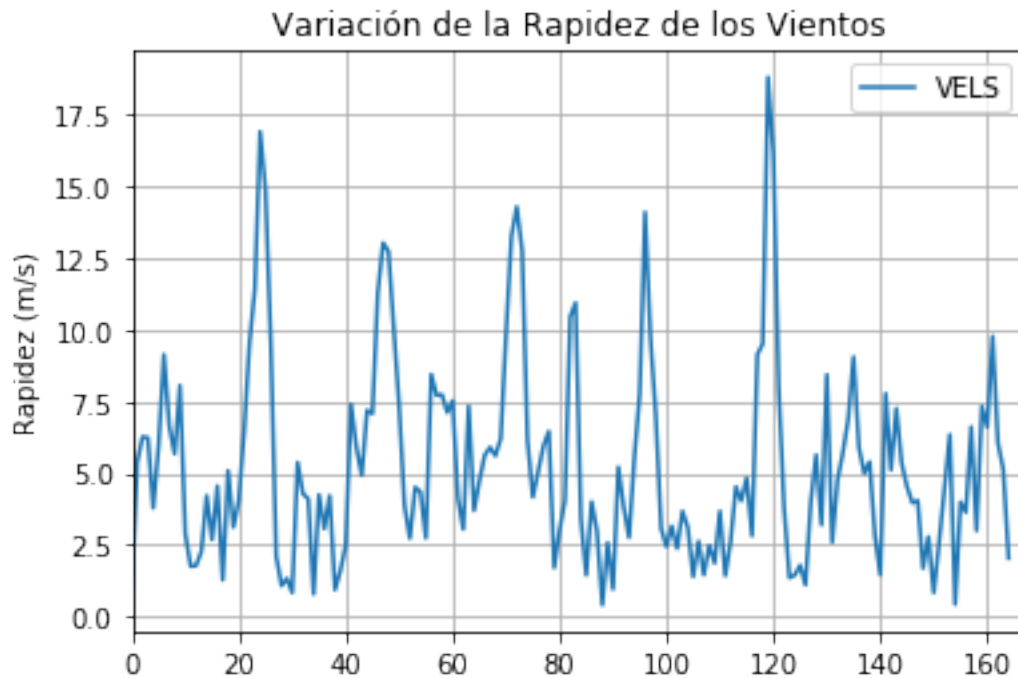
De igual manera a las partes de codigo anterior encontramos los comentarios que describen lo que se realizo en cada parte del mismo.

```
In [16]: # Gráfica de la rapidez de los vientos (m/s)
plt.figure(); df.VELR.plot(); plt.legend(loc='best')
plt.title("Variación de la Rapidez de los Vientos")
plt.ylabel("Rapidez (m/s)")
plt.grid(True)
plt.show()
```



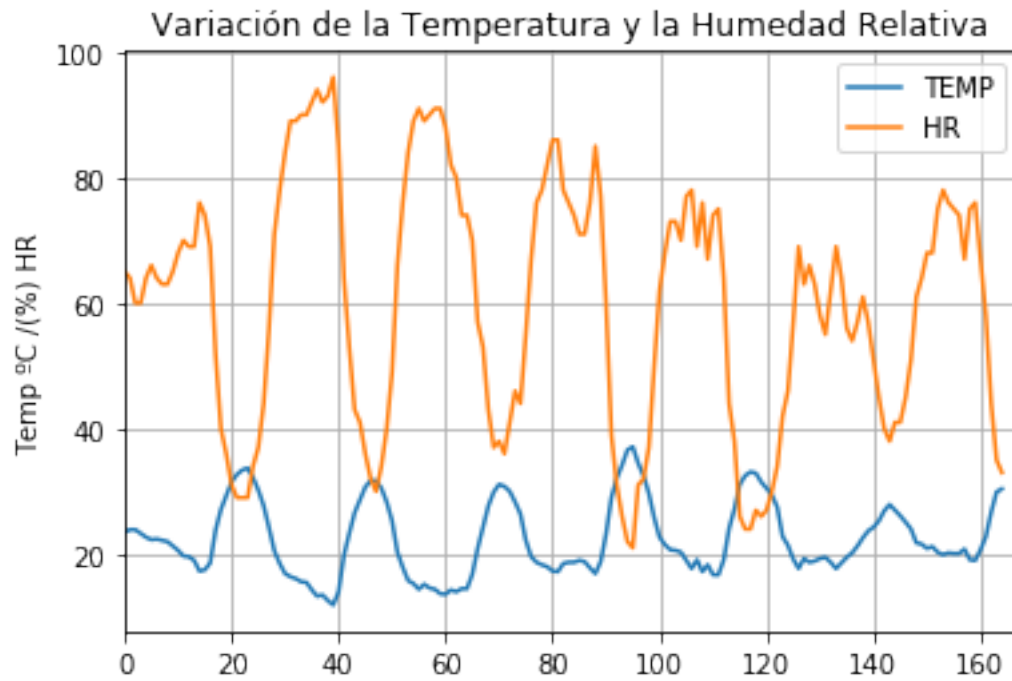
```
In [17]: # Gráfica de la rapidez de los vientos (m/s)
plt.figure(); df.VELS.plot(); plt.legend(loc='best')
plt.title("Variación de la Rapidez de los Vientos")
```

```
plt.ylabel("Rapidez (m/s)")
plt.grid(True)
plt.show()
```

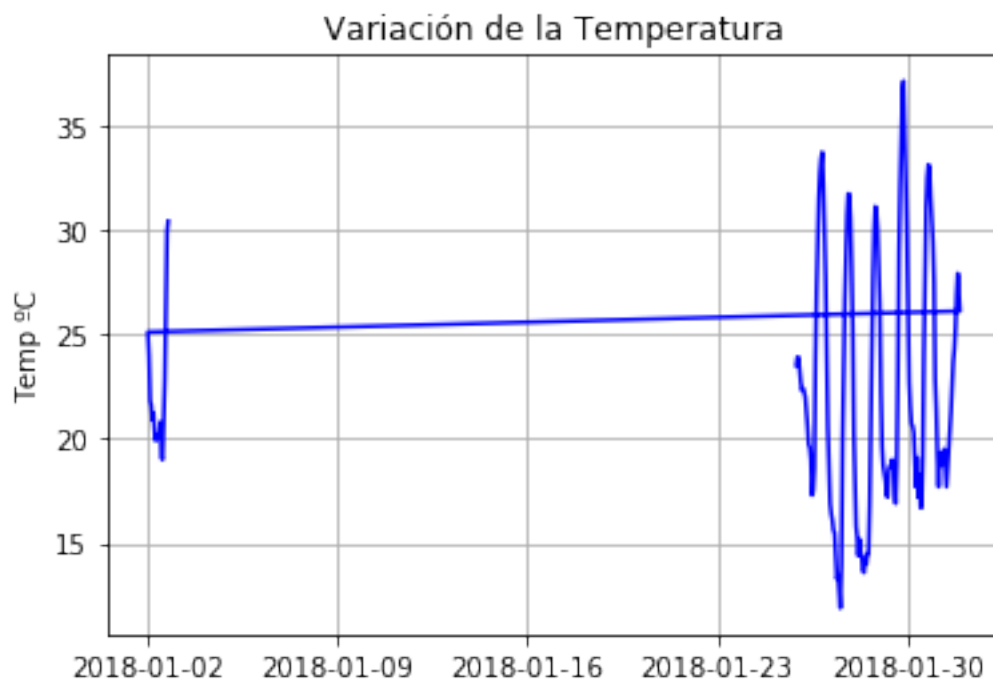


```
In [18]: # Gráfica de Temperatura y Humedad Relativa
df1 = df[['TEMP', 'HR']]
plt.figure(); df1.plot(); plt.legend(loc='best')
plt.title("Variación de la Temperatura y la Humedad Relativa")
plt.ylabel("Temp °C /(%) HR")
plt.grid(True)
plt.show()
```

```
<matplotlib.figure.Figure at 0x7f2549331048>
```



```
In [19]: plt.plot_date(x=df.FECHA, y=df.TEMP, fmt="b-")
plt.title("Variación de la Temperatura")
plt.ylabel("Temp °C")
plt.grid(True)
plt.show()
```

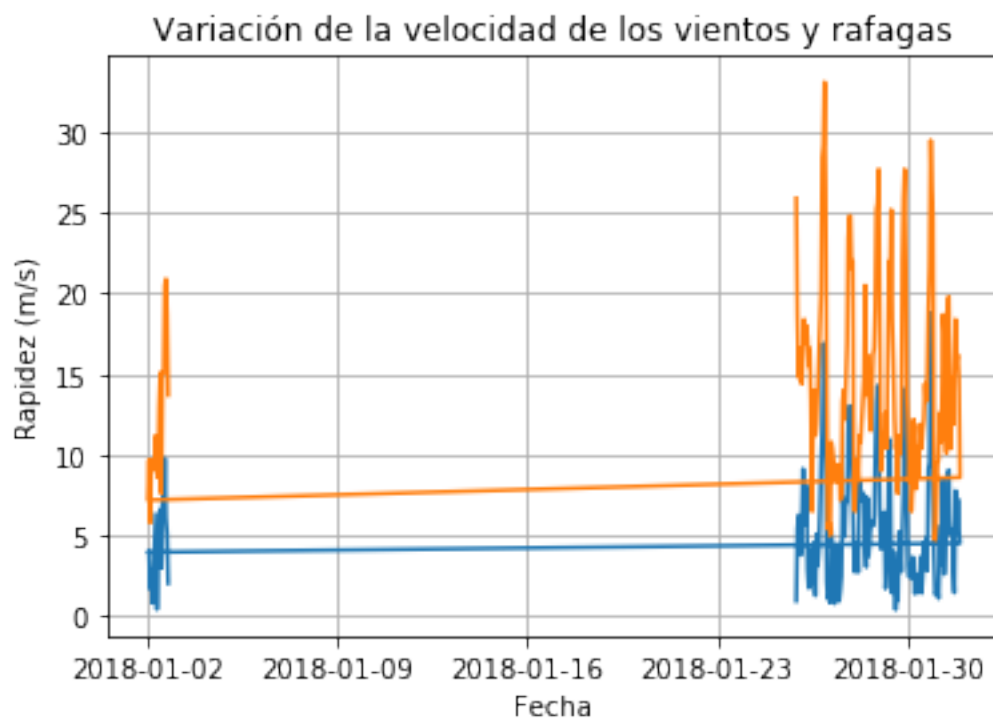




In [26]:

```
Out[26]: DIRS      160.038554
DIRR      144.200602
VELS       5.285636
VELR      14.347273
TEMP      22.446667
HR        59.993939
PB       1010.161212
PREC       0.000000
RAD-SOL    162.000000
dtype: float64
```

```
In [29]: # Gráfica de las velocidades de los vientos en funcion del tiempo
y=df[['VELS', 'VELR']]
x=df['FECHA']
plt.plot(x,y)
plt.title("Variación de la velocidad de los vientos y rafagas")
plt.ylabel("Rapidez (m/s)")
plt.xlabel("Fecha")
plt.grid(True)
plt.show()
```



```
In [30]: #Cambiar el nombre de una columna en los datos
df.rename(columns={'RAD-SOL': 'RADSOL'}, inplace=True)
```

```
In [31]: df.mean()
```

```
Out[31]: DIRS      160.038554
DIRR      144.200602
VELS       5.285636
VELR      14.347273
TEMP      22.446667
HR        59.993939
PB      1010.161212
PREC       0.000000
RADSOL     162.000000
dtype: float64
```

```
In [36]: # Gráfica de la radiación solar en función del tiempo
y=df[['RADSOL']]
x=df['FECHA']
plt.plot(x,y)
plt.title("Variación de la velocidad de los vientos y rafagas")
plt.ylabel("(w/m2).")
plt.xlabel("Fecha")
plt.grid(True)
plt.show()
```



Como ultima parte se utilizo el comando `df.describe()` el cual nos dio una descripción de los datos de manera mas estadística mostrandonos el promedio, la desviación, el minimo, el maximo y los cuantiles de cada una de las columnas de datos climaticos de la ciudad.

In [34]: `df.describe()`

```
Out [34]:
```

|       | DIRS        | DIRR       | VELS       | VELR       | TEMP \     |
|-------|-------------|------------|------------|------------|------------|
| count | 166.000000  | 166.000000 | 165.000000 | 165.000000 | 165.000000 |
| mean  | 160.038554  | 144.200602 | 5.285636   | 14.347273  | 22.446667  |
| std   | 107.303216  | 119.573799 | 3.509094   | 5.830790   | 5.855647   |
| min   | 29.000000   | 2.000000   | 0.410000   | 4.700000   | 11.900000  |
| 25%   | 77.750000   | 34.000000  | 2.760000   | 10.100000  | 18.500000  |
| 50%   | 142.500000  | 109.000000 | 4.520000   | 13.300000  | 20.900000  |
| 75%   | 233.750000  | 245.500000 | 7.030000   | 17.300000  | 27.000000  |
| max   | 1011.400000 | 424.300000 | 18.810000  | 33.100000  | 37.100000  |

|       | HR         | PB          | PREC  | RADSOL     |
|-------|------------|-------------|-------|------------|
| count | 165.000000 | 165.000000  | 165.0 | 165.000000 |
| mean  | 59.993939  | 1010.161212 | 0.0   | 162.000000 |
| std   | 19.709010  | 1.773710    | 0.0   | 245.541393 |
| min   | 21.000000  | 1006.100000 | 0.0   | -1.000000  |
| 25%   | 42.000000  | 1009.100000 | 0.0   | -1.000000  |
| 50%   | 64.000000  | 1010.100000 | 0.0   | -0.300000  |
| 75%   | 75.000000  | 1011.300000 | 0.0   | 278.000000 |
| max   | 96.000000  | 1014.300000 | 0.0   | 760.800000 |

### 1.3 Conclusiones

Al final de todo puedo terminar diciendo que actualmente la manera de trabajar con Python ha sido mas sencilla aun que sigo desconociendo bastantes comandos de el, pero, en general es mas amigable que los demas lenguajes, no tuve mucha dificultad y me impresiono mucho el hecho de poder graficar directamente con un comando sin necesidad de utilizar una aplicación diferente para hacerlo.

Me gusto mucho la experiencia dentro de este nuevo lenguaje, veo mucho potencial en su uso y esto podra ayudarme en otras ramas del conocimiento científico ademas de su gran potencial estadístico.

## 2 Apéndice

1. ¿Cuál es tu primera impresión de Jupyter Notebook?
  - Fue un entorno algo extraño, ya que nunca había programado en este tipo de paginas, ademas de que es un nuevo lenguaje.
2. ¿Se te dificultó leer código en Python?
  - Un poco ya que no estaba familiarizado, pero poco a poco iba aprendiendo a utilizarlo.
3. ¿En base a tu experiencia de programación en Fortran, que te parece el entorno de trabajar en Python?
  - Es muchisimo mas facil ya que la compilación es mas directa y rapida, ademas de que la forma de dar el codigo y comandos es muchisimo mas sencilla.
4. A diferencia de Fortran, ahora se producen las gráficas utilizando la biblioteca Matplotlib. ¿Cómo fue tu experiencia?
  - Genial, es una maravilla el poder gráfricar de manera tan rapida y sencilla.
5. En general, ¿qué te pareció el entorno de trabajo en Python?
  - Mas amigable a la vista, tiene un orden establecido muy agradable aun que se volvio algo confuso por ser ajeno a mi.
6. ¿Qué opinas de la actividad? ¿Estuvo compleja? ¿Mucho material nuevo? ¿Que le faltó o que le sobró? ¿Qué modificarías para mejorar?
  - Estuvo muy bien, utilizar datos climaticos creo que fue lo mejor ya que aun no estamos familiarizados con otro tipo de datos estadísticos dentro de la física con nuestro nivel actual, pero si fue algo muy nuevo para mi el tener que usar un nuevo lenguaje aun que el ejemplo proporcionado por el profesor me ayudo bastante.

CARLOSLIZARRAGAC/FISICACOMPUTACIONAL1 Bibliografía: GitHub.com (2018). carloslizarragac/FisicaComputacional1.com [online] Available at: <https://github.com/carloslizarragac/FisicaComputacional1> [Accessed 7 Feb. 2018].

PROJECT JUPYTER Bibliografía: Jupyter.org. (2018). Project Jupyter. [online] Available at: <http://jupyter.org/> [Accessed 7 Feb. 2018].

ESTACIONESAUTOMÁTICAS Bibliografía: Smn1.conagua.gob.mx. (2018). EstacionesAutomáticas. [online] Available at: <http://smn1.conagua.gob.mx/emas/> [Accessed 7 Feb. 2018].