



Nombre de la práctica	Manual de Practicas Matplotlib en Anaconda			No.	1
Asignatura:	Simulación 3501	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	8

I. Competencia(s) específica(s): **Alumno: Raúl Ciriaco Castillo 3501**

(Github:

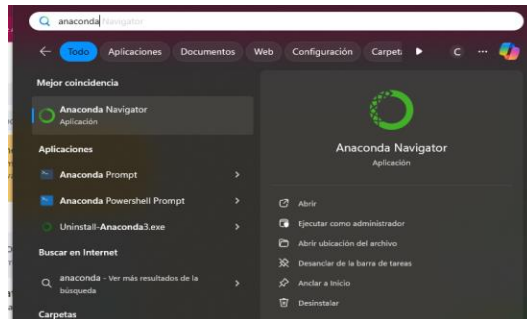
Comentado [T1]: <https://github.com/RaulCiriaco>

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

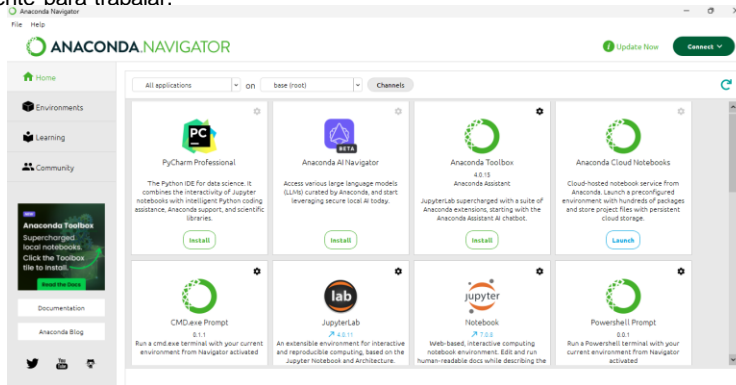
III. Material empleado:

Desarrollo de la práctica:

Como primeros pasos debemos instalar Anaconda Studio en nuestro Equipo de Computo que vamos a utilizar



Una vez que instalamos las paqueterías y todo lo requerido abrimos Anaconda y procedemos a crear nuestro ambiente para trabajar.

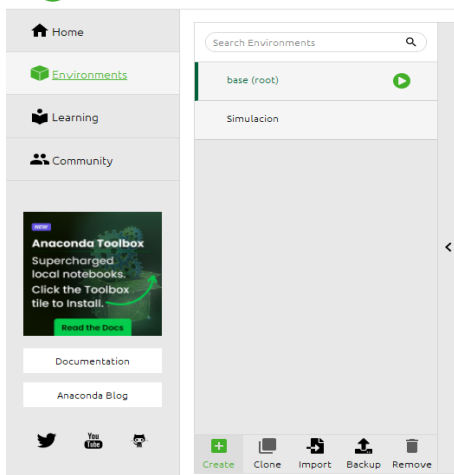


MANUAL DE PRÁCTICAS



File Help

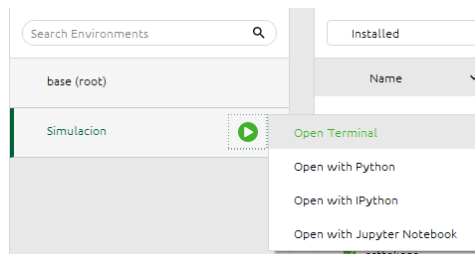
ANACONDA.NAVIGATOR



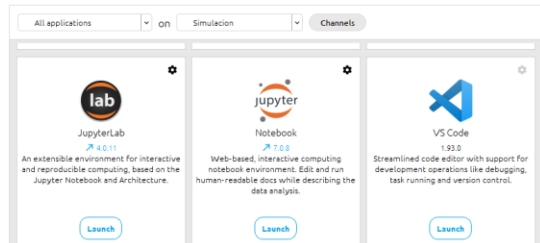
Debemos dar click en el apartado “Environments” y abajo pulsamos el “Create”

Después pulsamos en Open Terminal y vamos a proceder a instalar las paqueterías:

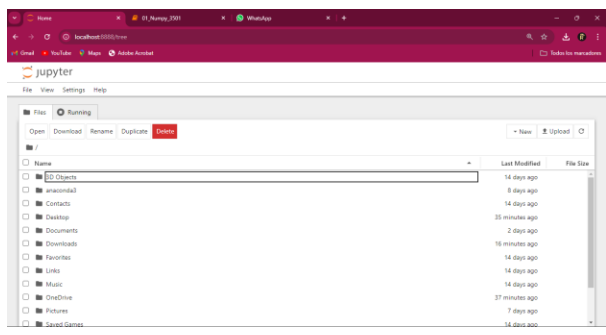
- Numpy
- Pandas
- Matplotlib
- SkLearn



Vamos a descargar el Jupyter Notebook y pulsamos en Launch.



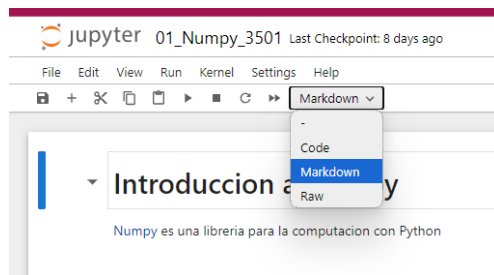
Nos abrirá esta pantalla donde podremos navegar libremente, crear archivos, modificar y eliminarlos.





Crearemos un nuevo archivo para comenzar con la librería “Matplotlib”.

En este apartado principal tendremos las herramientas que podemos utilizar en nuestro archivo, tenemos las opciones que se muestran en la imagen



- **Code:** Este apartado nos sirve para poder realizar instrucciones que queramos ejecutarlas después, que cumplan con alguna condición o no, cualquier actividad que tengamos que codificar se verá reflejada en este espacio.
- **Markdown:** Este nos permite la escritura de cualquier información, ya sean formulas o códigos para ejemplificar, no funciona si queremos mostrar o imprimir algún dato o valor.

Comenzaremos con una línea que nos permitirá enlazar con la página principal de la librería, en la cual podemos acceder para conocer información acerca de la misma, cualquier duda que tengamos podemos encontrarla en la página oficial.

Introduccion a Matplotlib

Matplotlib Es una biblioteca que permite la creacion de figuras y graficas de calidad mediante el uso de Python.

- Permite la creación de graficos de manera sencilla y eficiente.
- Permite la integración de graficos y figuras en Jupyter Notebook.

Para comenzar a trabajar, necesariamente debemos importar la paquetería de Numpy.

Import

```
[2]: import matplotlib
import matplotlib.pyplot as plt

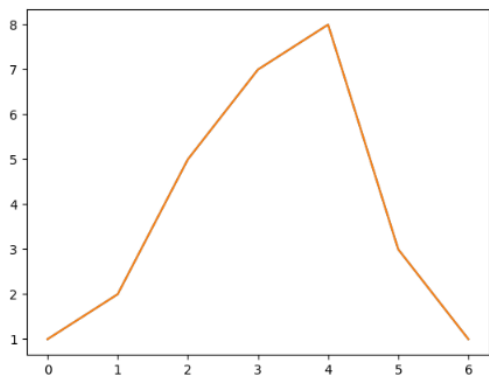
[3]: # Muestra los graficos integrados dentro de Jupyter Notebook
%matplotlib inline
```



Representación gráfica de Datos.

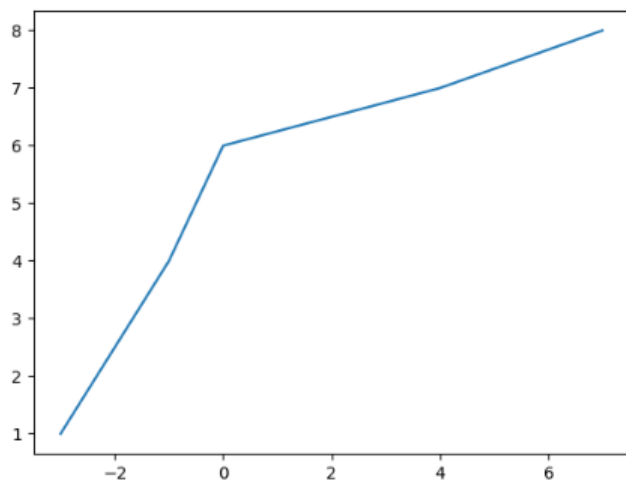
Si a la función de trazado se le da una matriz de datos, la usará como coordenadas en eje vertical, y utilizará el índice de cada punto de datos en el array, como la coordenada horizontal.

```
[ ]: plt.plot([1, 2, 5, 7, 8, 3, 1])  
plt.show()
```



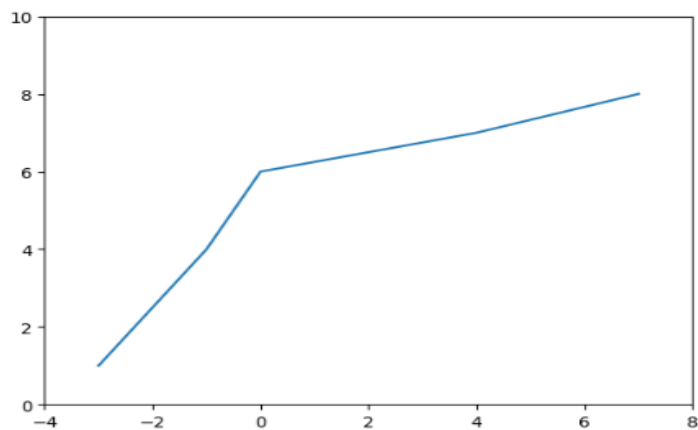
También se pueden proporcionar dos matrices: una para el eje horizontal y otra para el eje vertical.

```
[9]: plt.plot([-3, -1, 0, 4, 7], [1, 4, 6, 7, 8])  
plt.show()
```



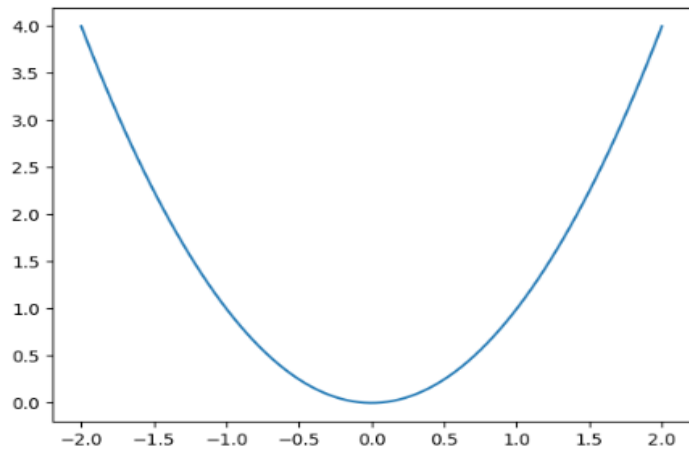


```
[10]: plt.plot([-3, -1, 0, 4, 7], [1, 4, 6, 7, 8])  
plt.axis([-4, 8, 0, 10]) #[xmin, xmax, ymin, ymax]  
plt.show()
```



Se sigue el mismo procedimiento para pintar una función matemática

```
[12]: import numpy as np  
x = np.linspace(-2, 2, 500)  
y = x**2  
plt.plot(x, y)  
plt.show()
```

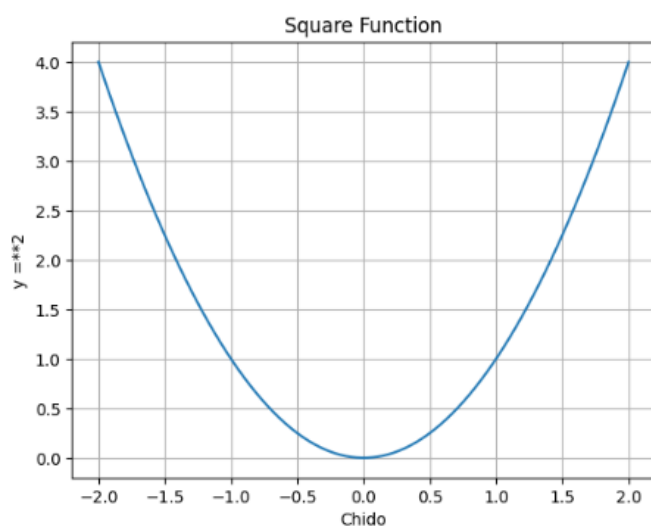


MANUAL DE PRÁCTICAS



También puede modificarse el estilo de la gráfica para que contenga más información.

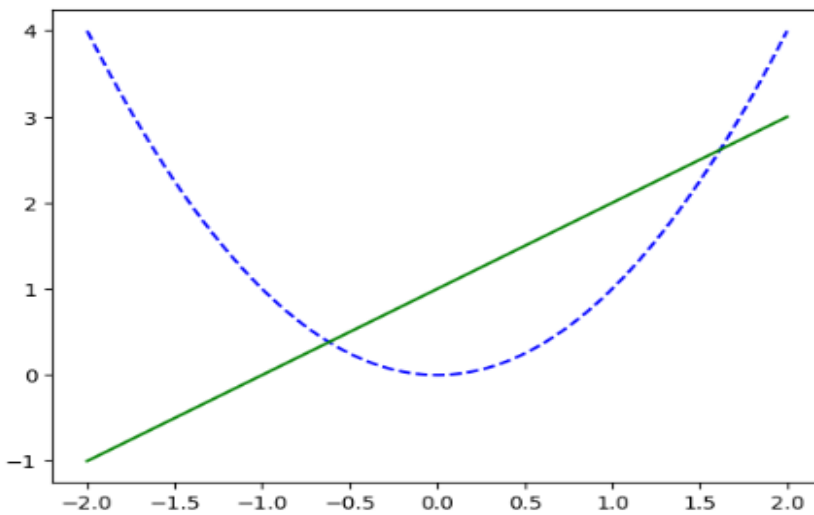
```
[13]: plt.plot(x, y)
plt.title("Square Function")
plt.xlabel("Chido")
plt.ylabel("y == x^2")
plt.grid(True)
plt.show()
```





Pueden superponers graficas y cambiar el estilo de las funciones.

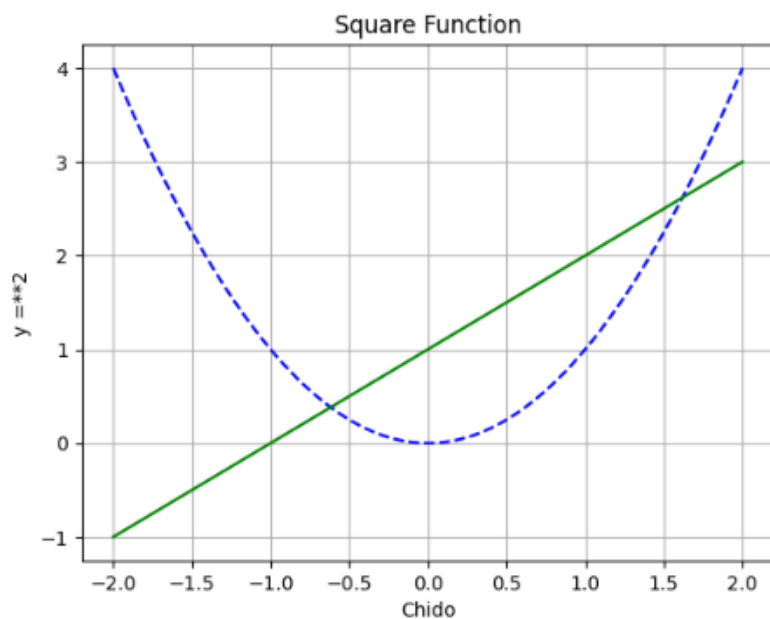
```
[15]: import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2
y2 = x + 1
plt.plot(x, y, 'b--', x, y2, 'g')
plt.show()
```



```
[16]: # Separando en diferentes lineas Las funciones
import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2
y2 = x + 1

plt.title("Square Function")
plt.xlabel("Chido")
plt.ylabel("y = **2")
plt.grid(True)

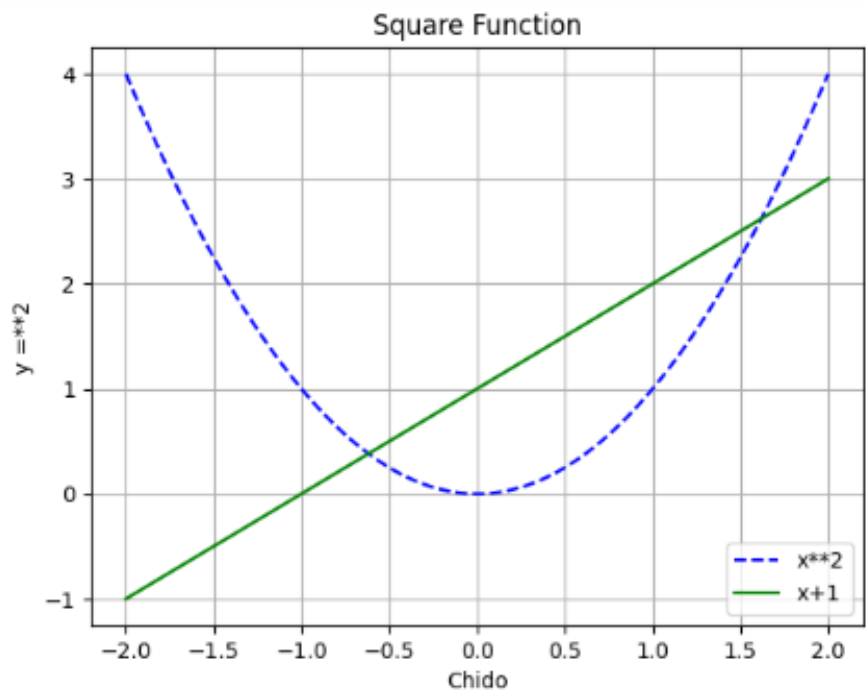
plt.plot(x, y, 'b--')
plt.plot(x, y2, 'g')
plt.show()
```



```
[25]: import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2
y2 = x + 1

plt.title("Square Function")
plt.xlabel("Chido")
plt.ylabel("y = x**2")
plt.grid(True)

plt.plot(x, y, 'b--', label="x**2")
plt.plot(x, y2, 'g', label="x+1")
plt.legend(loc = "best") #La situa en la mejor Localizacion
plt.show()
```

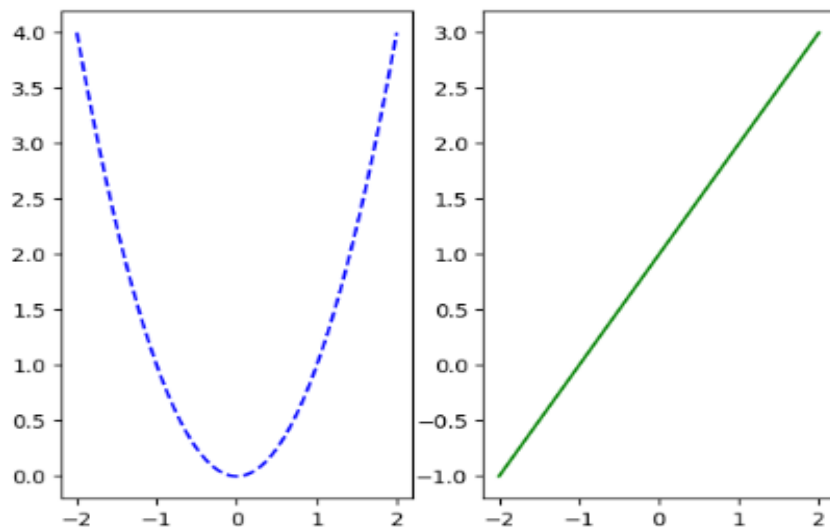
También puede crearse dos gráficas que no se superpongan, estas gráficas se organizan en un grid y se denominan subplots

```
import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2
y2 = x + 1

plt.subplot(1, 2, 1) # 1 Rows, 2 Columns, 1st SubPlot
plt.plot(x, y, 'b--')

plt.subplot(1, 2, 2) # 1 Rows, 2 Columns, 2nd SubPlot
plt.plot(x, y2, 'g')

plt.show()
```



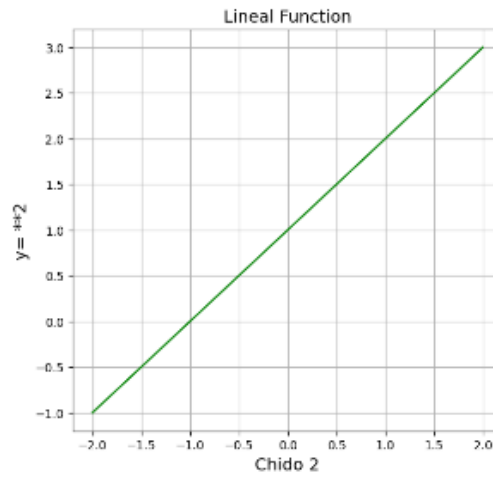
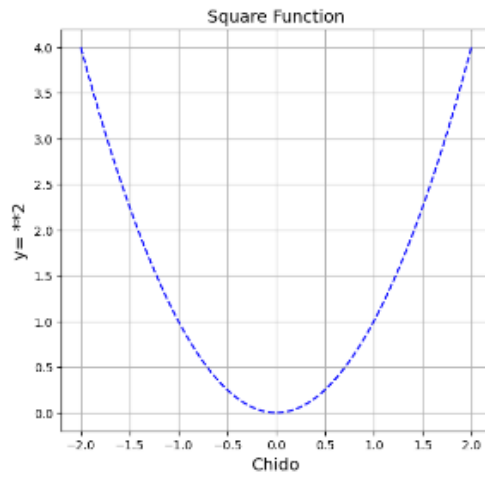
Para que las graficas no queden tan ajustadas, se pueden hacer la figura mas grande.

```
[31]: plt.figure(figsize =(14, 6))

plt.subplot(1, 2, 1) # 1 Rows, 2 Columns, 1st Subplot
plt.plot(x, y, 'b--')
plt.title("Square Function", fontsize = 14)
plt.xlabel("Chido", fontsize = 14)
plt.ylabel("y= **2", fontsize = 14)
plt.grid(True)

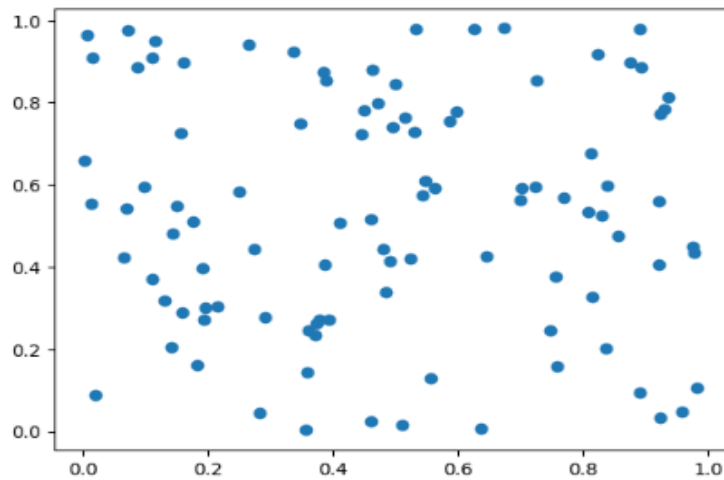
plt.subplot(1, 2, 2) # 1 Rows, 2 Columns, 2nd Subplot
plt.plot(x, y2, 'g')
plt.title("Lineal Function", fontsize = 14)
plt.xlabel("Chido 2", fontsize = 14)
plt.ylabel("y= **2", fontsize = 14)
plt.grid(True)

plt.show()
```



Scatter Plots

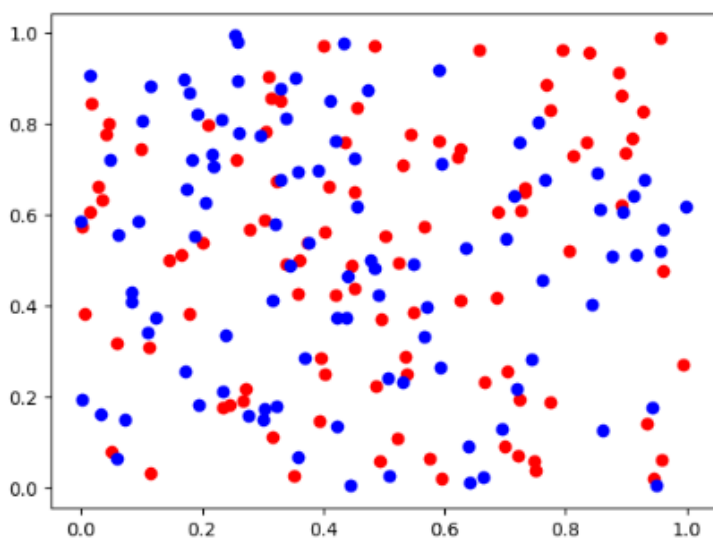
```
[32]: from numpy.random import rand
x, y = rand(2, 100)
plt.scatter(x, y)
plt.show()
```





```
[34]: from numpy.random import rand
x, y = rand(2, 100)
x2, y2 = rand(2, 100)
plt.scatter(x, y, c= 'red')
plt.scatter(x2, y2, c= 'blue')

plt.show()
```



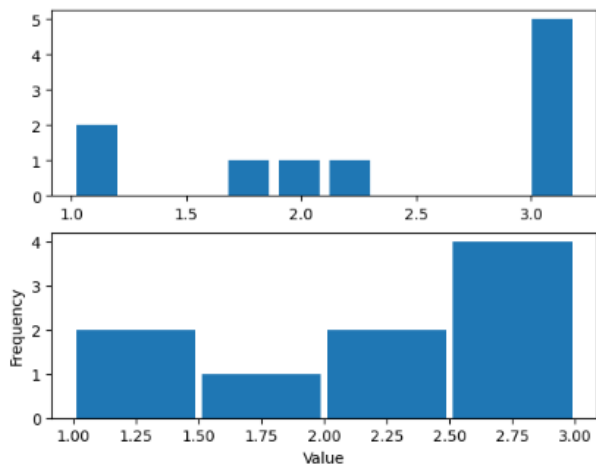
▼ Histogramas ¶

```
9]: data = [1, 1.1, 1.8, 2, 2.1, 3.2, 3, 3, 3, 3]
plt.subplot(211)
plt.hist(data, bins = 10, rwidth= 0.8)

plt.subplot(212)
plt.hist(data, bins = [1, 1.5, 2, 2.5, 3], rwidth = 0.95)
plt.xlabel('Value')
plt.ylabel('Frequency')

plt.show()
```

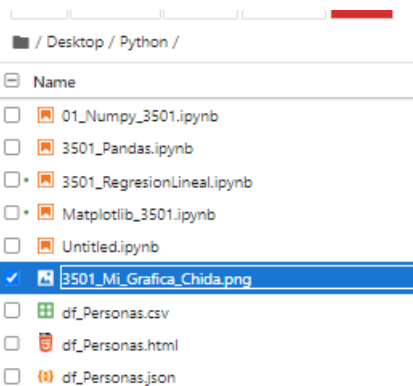
MANUAL DE PRÁCTICAS



Guardar las figuras

```
from numpy.random import rand
x, y = rand(2, 100)
x2, y2 = rand(2, 100)
plt.scatter (x, y, c = 'red')
plt.scatter (x2, y2, c = 'blue')

plt.savefig("3501_Mi_Grafica_Chida.png", transparent = True)
```





IV. Conclusiones:

La verdad este tema que vimos durante las clases nos ayudó a fortalecer mucho nuestro conocimiento sobre Anaconda, trabajando con esta nueva librería Matplotlib que prácticamente se basa en el diseño gráfico de funciones, con la representación de datos y funciones mediante graficas que nos permite conocer el movimiento de los datos y de igual manera recopilar cabe recalcar que semestres pasados el docente aprovechaba un rato para impartirnos este conocimiento en clase y eso fue clave fundamental para poder facilitar este trabajo en el semestre presente, sinceramente esperamos seguir trabajando de esta manera con el docente ya que es una forma muy manejable y muy buena para entender y comprender las clases, algo ameno que como estudiantes hace que podamos comprender cada tema y así aprovechar cualquier momento para preguntar al maestro si en algún momento tenemos dudas sobre el tema.

El uso de las herramientas que nos proporciona el docente para trabajar facilitan la comprensión del tema en su extensión, mediante Anaconda Navigator y el uso de distintas librerías podemos analizar y comprender el movimiento sobre los datos que queramos analizar, todo eso con el fin de recopilar información útil para cada situación.

