Nombre de la práctica	MATPLOTLIB			No.	3
Asignatura:	Simulación	Carrera:	Inf. Sistemas Computacionales	Duración de la práctica (Hrs)	

Nombre: Ana Edith Hernández Hernández

I. Competencia(s) específica(s):

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

III. Material empleado:

Laptop
Anaconda

IV. Desarrollo de la práctica:

Matplotlib.

Comenzaremos con una introducción.
Hacemos la importación.

Introducción a Matplotlib

Matplotlib Es una biblioteca que permite la creación de figuras y gráficos de calidad mediante el uspo de Python.

- Permite la creación de gráficosd e manera sencilla y eficiente.
- Permite la integración de gráficos y figuras en un Jupyter Notebook.

Import

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

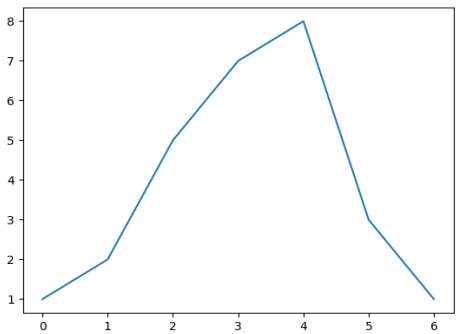
```
[2]: # Muestra los gráficos integrados dentro de Jupyter Notebook
     %matplotlib inline
```

Hacemos la representación gráfica.

Representación gráfica de Datos.

Si a la funciuion de trazadon se le da una matriz de datos, la usara como coordenadas en eje vertical, y utilizara en índic de cada punto de datos en el array como la coordenada horizontal

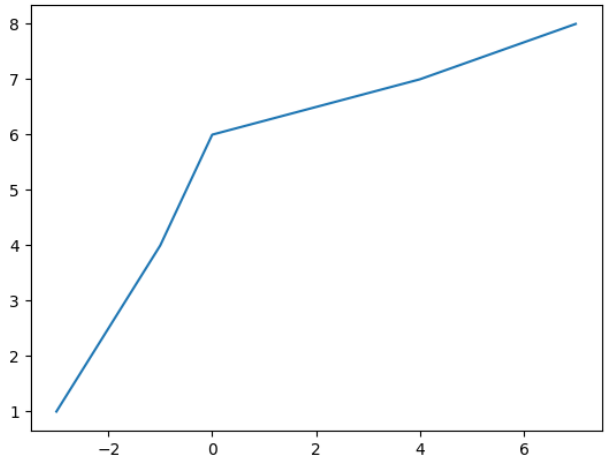
```
[4]: plt.plot([1, 2, 5, 7, 8, 3, 1])
     plt.show()
```



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

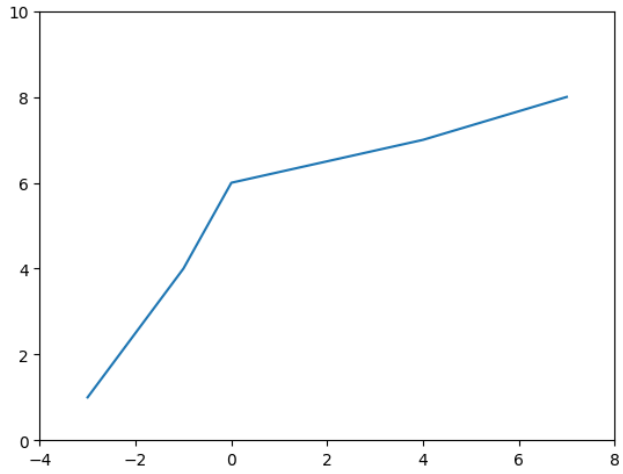
```
[5]: plt.plot([-3, -1, 0, 4, 7], [1, 4, 6, 7, 8])
```

```
[5]: [<matplotlib.lines.Line2D at 0x7186901496d0>]
```



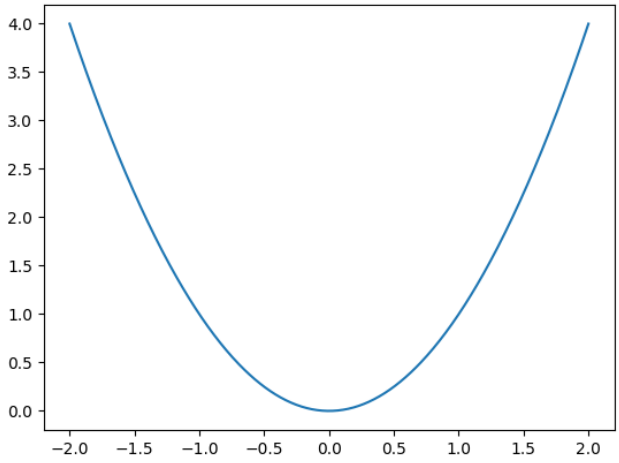
Pueden modificarae las longitudes de los ejes para que la figura no se vea tan ajustada

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



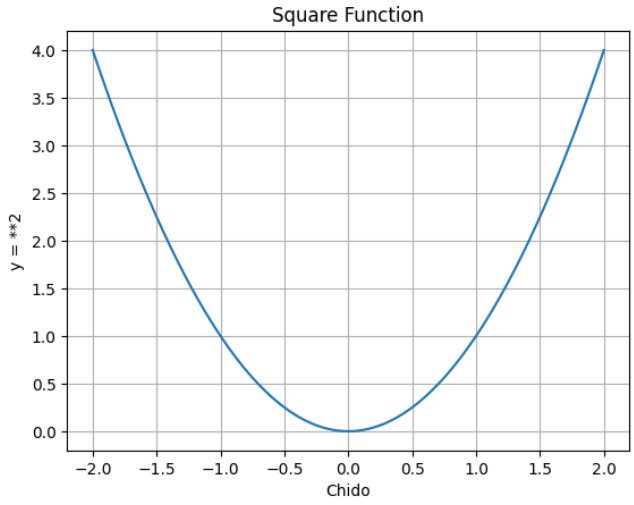
Se sigue el mismo procedimiento para pintar una función matemática

[8]:
```
import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2

plt.plot(x, y)
plt.show()
```



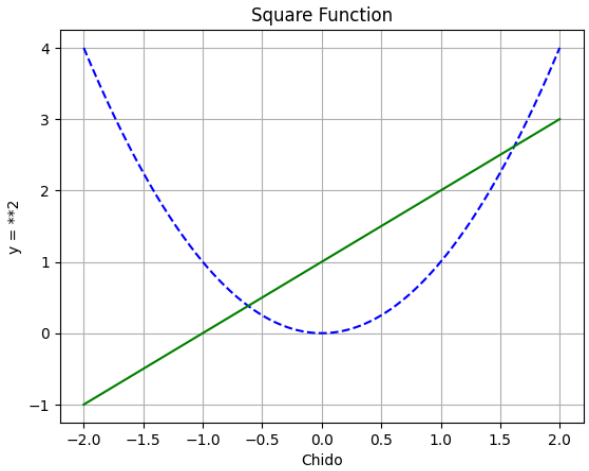
También puede modificarse el estilo de la gráfica para que contenga mas información.

[11]:
```
plt.plot(x, y)
plt.title("Square Function")
plt.xlabel("Chido")
plt.ylabel("y = **2")
plt.grid(True)
plt.show()
```

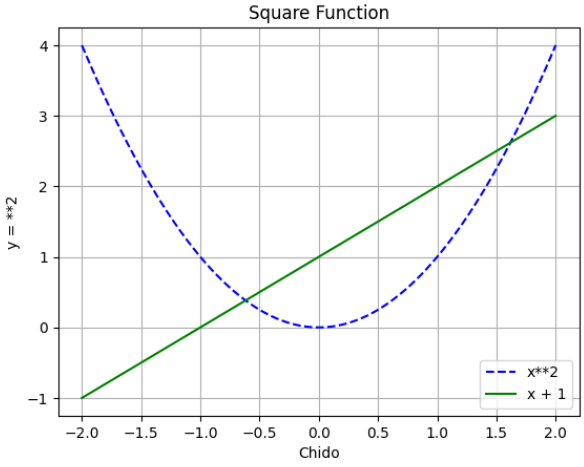


Pueden superponece gráficas y cambiar el estilo de las funciones

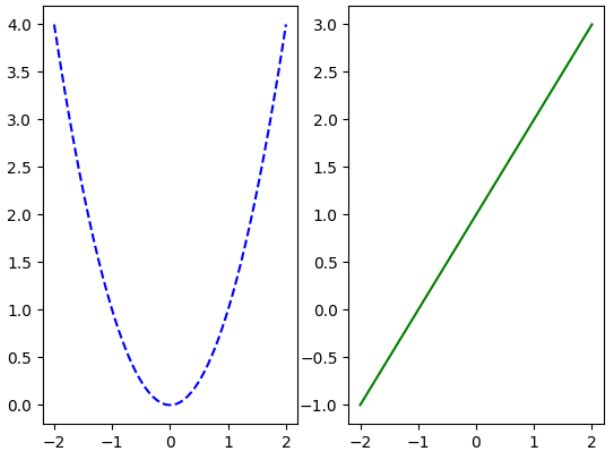
[16]:
```
# Separando en diferentes lineas las funciones.
import numpy as np
x = np.linspace(-2, 2, 500)
y = x**2
y2 = x + 1


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()
```



[18]:
```
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--', label = "x**2")
plt.plot(x, y2, 'g', label = "x + 1")
plt. legend(loc = "best") # La situa en la mejor localizacion
plt.show()
```

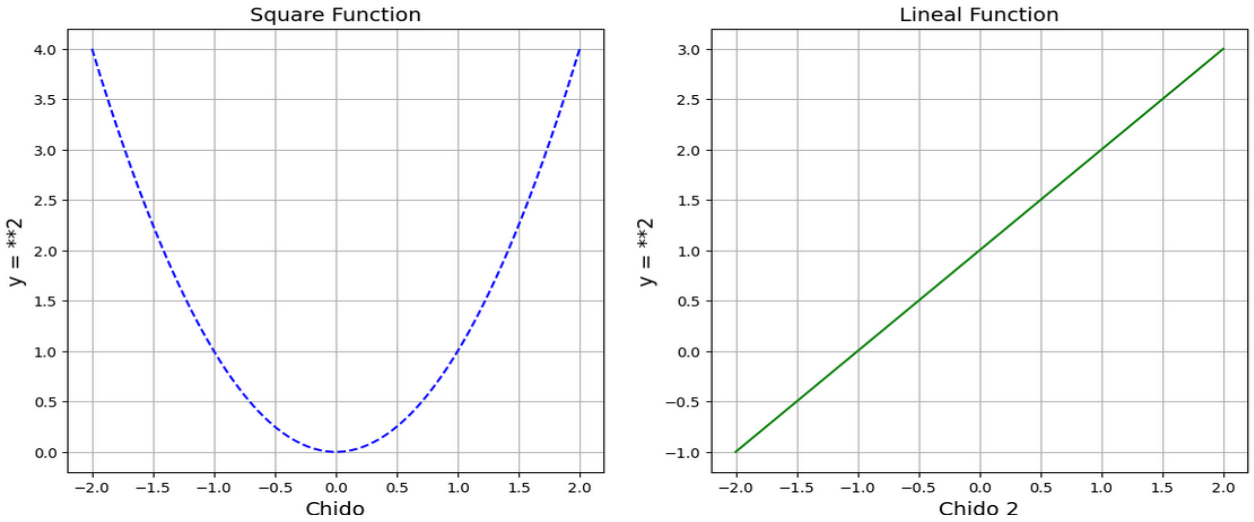


[21]:
```
import numpy as np
x = np.linspace (-2, 2, 500)
y = x**2
y2 = x+1

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 Suplot
plt.plot(x, y2, 'g')

plt.show()
```



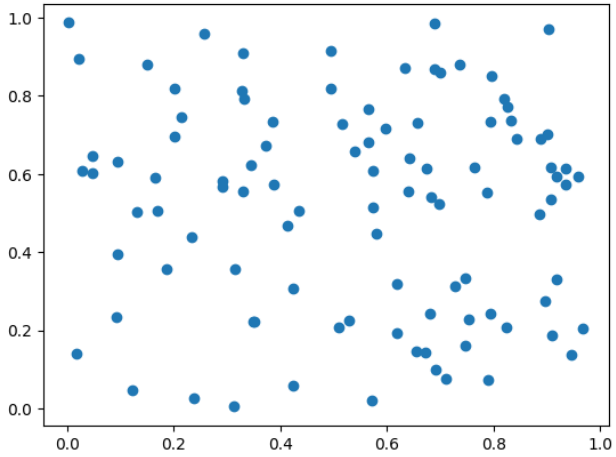
Para que las graficas no queden tn ajustadas, se puede hacer la figura mas grande

[25]:
```
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 Suplot
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()
```

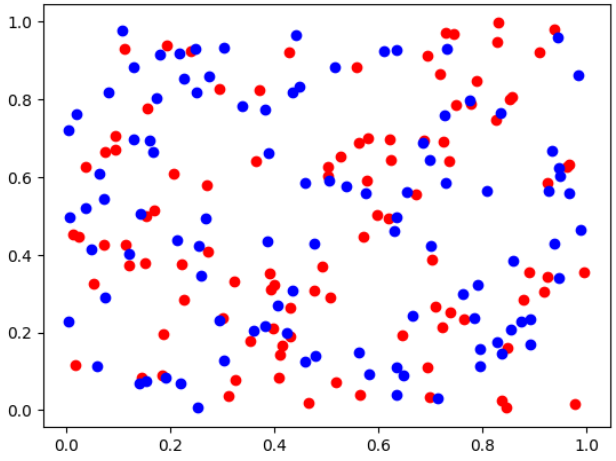


Scatter Plots

[26]:
```
from numpy.random import rand
x, y = rand(2, 100)
plt.scatter(x, y)
plt.show()
```

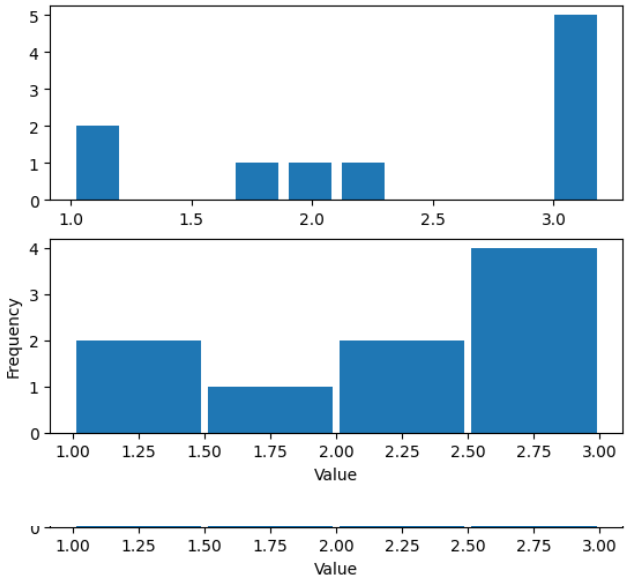


[27]:
```
from numpy.random import rand
x, y = rand(2, 100)
x2, y2 = rand(2, 100)
plt.scatter(x, y, c = 'red')
plt.scatter(x2, y2, c = 'blue')
plt.show()
```



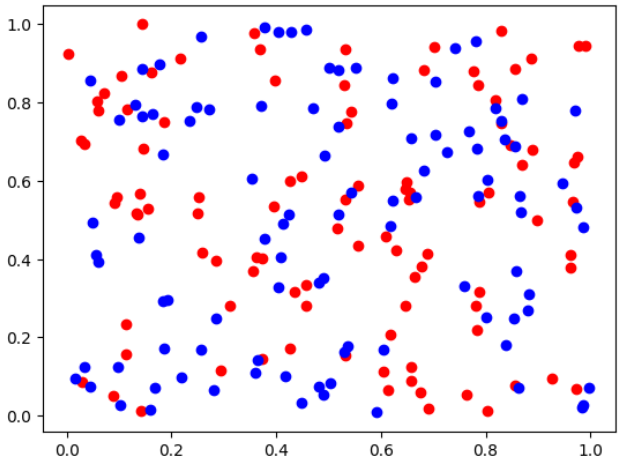
Histogramas

[30]:
```
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()
```



Guardar las fuguras

[33]:
```
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)
```



Conclusiones:

El uso de matplotlib es fundamental para poder hacer la creación de gráficos en Python.

Este principalmente nos ayuda a hacer el diseño de estos gráficos, desde poner el nombre al pie de una gráfica de barras hasta el color delas barras, esto de manera sencilla. Al igual la creación de figuras que representen algún tipo de datos.