Programación Científica y HPCI

Máster Universitario en Ingeniería Matemática y Computación

Tema 6

unir LA UNIVERSIDAD EN INTERNET

Universidad Internacional de La Rioja

¿Cómo estudiar este tema?

Material de Estudio

Tema 6

Material Complementario

Preparar el entorno para la programación científica. Anaconda

Tutorial de NumPy

Creación de paneles en Pandas

NumPy y SciPy

Visualizaciones gráficas con `matplotlib`

Lección magistral

Numpy

- **Tipo de datos ndarray**

```
In [47]: x=np.array([1,2,3])

In [48]: x
Out[48]: array([1, 2, 3])

In [49]: x[0]
Out[49]: 1
```

```
In [51]: y=np.arange(3)

In [52]: y
Out[52]: array([0, 1, 2])

In [53]: y[0]
Out[53]: 0
```

import numpy as np

```
In [56]: z=np.linspace(0,3)

In [57]: z
Out[57]:
array([0.        , 0.06122449, 0.12244898, 0.18367347, 0.24489796,
       0.30612245, 0.36734694, 0.42857143, 0.48979592, 0.55102041,
       0.6122449 , 0.67346939, 0.73469388, 0.79591837, 0.85714286,
       0.91836735, 0.97959184, 1.04081633, 1.10204082, 1.16326531,
       1.2244898 , 1.28571429, 1.34693878, 1.40816327, 1.46938776,
       1.53061224, 1.59183673, 1.65306122, 1.71428571, 1.7755102 ,
       1.83673469, 1.89795918, 1.95918367, 2.02040816, 2.08163265,
       2.14285714, 2.20408163, 2.26530612, 2.32653061, 2.3877551 ,
       2.44897959, 2.51020408, 2.57142857, 2.63265306, 2.69387755,
       2.75510204, 2.81632653, 2.87755102, 2.93877551, 3.        ])

In [58]: len(z)
Out[58]: 50

In [59]: z[48]
Out[59]: 2.9387755102040813
```

```
In [60]: t=np.logspace(0,3)

In [61]: t
Out[61]:
array([   1.        ,    1.1513954 ,    1.32571137,    1.52641797,
          1.75751062,    2.02358965,    2.32995181,    2.6826958 ,
          3.0888436 ,    3.55648031,    4.09491506,    4.71486636,
          5.42867544,    6.25055193,    7.19685673,    8.28642773,
          9.54095476,   10.98541142,   12.64855217,   14.56348478,
         16.76832937,   19.30697729,   22.22996483,   25.59547923,
         29.47051703,   33.93221772,   39.06939937,   44.98432669,
         51.79474679,   59.63623317,   68.6648845 ,   79.06043211,
         91.0298178 ,  104.81131342,  120.67926406,  138.94954944,
        159.98587196,  184.20699693,  212.09508879,  244.20530945,
        281.1768698 ,  323.74575428,  372.75937203,  429.19342601,
        494.17133613,  568.9866029 ,  655.12855686,  754.31200634,
        868.51137375, 1000.        ])

In [62]: len(t)
Out[62]: 50
```

Numpy. Algebra lineal

```
In [78]: import numpy as np
    ...:
    ...: A = [[1, 3], [2, 1]]
    ...: B = [[1, 1], [3, 1]]
    ...: AxB = np.dot(A, B)
    ...:
    ...: from numpy import linalg
    ...: detA=int(np.linalg.det(A))
    ...: detB=np.linalg.det(B)

In [79]: A
Out[79]: [[1, 3], [2, 1]]

In [80]: B
Out[80]: [[1, 1], [3, 1]]

In [81]: AxB
Out[81]:
array([[10,  4],
       [ 5,  3]])

In [82]: detA
Out[82]: -5

In [83]: detB
Out[83]: -2.0000000000000004
```

Vectorización

Permite la aplicación de una función definida para escalares sobre un vector

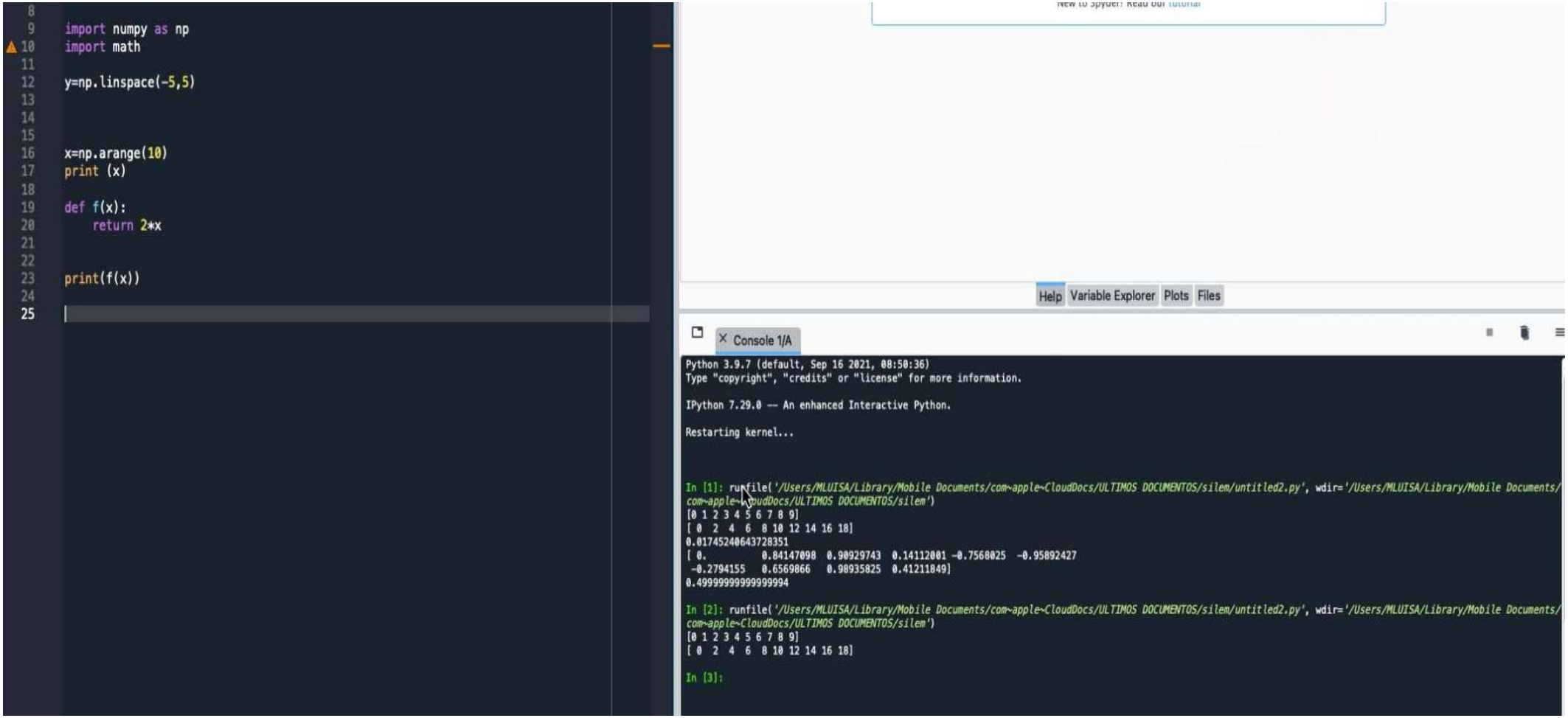
```
In [74]: x=np.arange(10)

In [75]: x
Out[75]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [76]: sen_vec=np.vectorize(math.sin)

In [77]: sen_vec(x)
Out[77]:
array([ 0.        ,  0.84147098,  0.90929743,  0.14112001, -0.7568025 ,
       -0.95892427, -0.2794155 ,  0.6569866 ,  0.98935825,  0.41211849])
```

expresión algebraica vs función


import numpy as np
import math
y=np.linspace(-5,5)
x=np.arange(10)
print (x)
def f(x):
return 2*x
print(f(x))
Help
Variable Explorer
Plots
Files
Console 1/A
Python 3.9.7 (default, Sep 16 2021, 08:50:36)
Type "copyright", "credits" or "license" for more information.
IPython 7.29.0 -- An enhanced Interactive Python.
Restarting kernel...
In [1]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[0 2 4 6 8 10 12 14 16 18]
0.01745240643728351
[0. 0.84147098 0.90929743 0.14112001 -0.7568025 -0.95892427
-0.2794155 0.6569866 0.98935825 0.41211849]
0.49999999999999994
In [2]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[0 2 4 6 8 10 12 14 16 18]
In [3]:

```python
import numpy as np
import math

y=np.linspace(-5,5)



x=np.arange(10)
print (x)

def f(x):
    return 2*x


print(f(x))

print(math.sin(x))


sen_vec=np.vectorize(math.sin)

print(sen_vec(x))
```

Help | Variable Explorer | Plots | Files

Console 1/A

```
[ 0.          0.84147098  0.90929743  0.14112001 -0.7568025  -0.95892427
 -0.2794155   0.6569866   0.98935825  0.41211849]
0.49999999999999994

In [2]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]

In [3]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]
Traceback (most recent call last):

  File "/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py", line 25, in <module>
    print(math.sin(x))

TypeError: only size-1 arrays can be converted to Python scalars


In [4]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]
Traceback (most recent call last):

  File "/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py", line 25, in <module>
    print(math.sin(x))

TypeError: only size-1 arrays can be converted to Python scalars
```

```python
import numpy as np
import math

y=np.linspace(-5,5)



x=np.arange(10)
print (x)

def f(x):
    return 2*x


print(f(x))


sen_vec=np.vectorize(math.sin)

print(sen_vec(x))
```

Help | Variable Explorer | Plots | Files

Console 1/A

```
[ 0  2  4  6  8 10 12 14 16 18]

In [3]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLU
com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]
Traceback (most recent call last):

  File "/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py", line 25, in <module>
    print(math.sin(x))

TypeError: only size-1 arrays can be converted to Python scalars


In [4]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLU
com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]
Traceback (most recent call last):

  File "/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py", line 25, in <module>
    print(math.sin(x))

TypeError: only size-1 arrays can be converted to Python scalars


In [5]: runfile('/Users/MLUISA/Library/Mobile Documents/com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem/untitled2.py', wdir='/Users/MLU
com~apple~CloudDocs/ULTIMOS DOCUMENTOS/silem')
[0 1 2 3 4 5 6 7 8 9]
[ 0  2  4  6  8 10 12 14 16 18]
[ 0.          0.84147098  0.90929743  0.14112001 -0.7568025  -0.95892427
 -0.2794155   0.6569866   0.98935825  0.41211849]
```

Scipy

Biblioteca para problemas matemáticos, científicos e ingeniería

Módulo	Descripción
scipy.io	Módulos, clases y funciones disponibles para leer y escribir datos en una variedad de formatos de archivo.
scipy.special	Funciones universales para funciones de Bessel, función elíptica, función gamma y funciones para el manejo de errores
scipy.linalg	Funciones para operaciones de álgebra lineal
scipy.interpolate	Funciones para interpolación
scipy.optimize	Funciones para optimización de funciones objetivo, posiblemente sujetas a restricciones. Incluye solucionadores para problemas no lineales (con soporte para algoritmos de optimización local y global), programación lineal, mínimos cuadrados restringidos y no lineales, búsqueda de raíces y ajuste de curvas.
scipy.stats	Biblioteca de funciones estadísticas con un gran número de distribuciones de probabilidad definidas.
scipy.integrate	Biblioteca para integración numérica y ecuaciones diferenciales.
scipy.ftt	Funciones para transformadas discretas de Fourier.
scipy.signal	Funciones para el procesamiento de señales.
scipy.ndimage	Funciones para el procesamiento de imágenes multidimensionales.

Scipy. linalg

```
In [1]: from scipy import linalg

In [2]: import numpy as np

In [3]: A=np.array([[1,2],[2,4]])

In [4]: B=np.array([[1,3],[2,1]])

In [5]: A
Out[5]:
array([[1, 2],
       [2, 4]])

In [6]: B
Out[6]:
array([[1, 3],
       [2, 1]])

In [7]: detA=linalg.det(A)

In [8]: detA
Out[8]: 0.0

In [9]: detB=linalg.det(B)

In [10]: detB
Out[10]: -5.0

In [11]: inv_A=linalg.inv(A)
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
<ipython-input-11-ff0bbf6cb19e> in <module>
----> 1 inv_A=linalg.inv(A)

/opt/anaconda3/envs/Pandas/lib/python3.8/site-packages/scipy/linalg/basic.py in inv(a, overwrite_a, check_finite)
    961         inv_a, info = getri(lu, piv, lwork=lwork, overwrite_lu=1)
    962     if info > 0:
--> 963         raise LinAlgError("singular matrix")
    964     if info < 0:
    965         raise ValueError('illegal value in %d-th argument of internal '

LinAlgError: singular matrix

In [12]: inv_B=linalg.inv(B)

In [13]: inv_B
Out[13]:
array([[-0.2,  0.6],
       [ 0.4, -0.2]])
```

Matplotlib

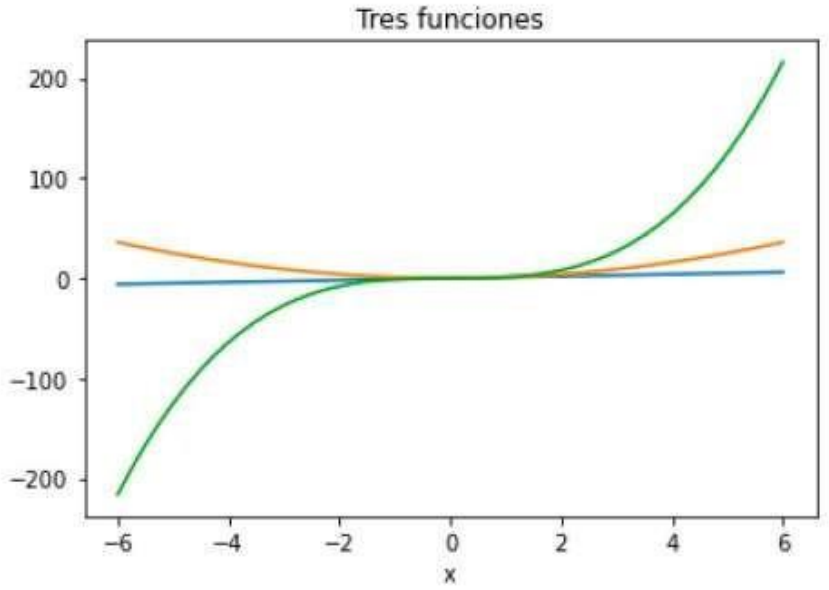
Biblioteca para es una completa biblioteca para crear visualizaciones estáticas, animadas e interactivas en Python.

Es importante distinguir entre los Axes o áreas de dibujo contenidas en una figura para mostrar las gráficas y los ejes X e Y (axis)

Función	Descripción
figure(numeración,tamaño,resolución,color del fondo, color del perímetros,valor lógico para mostrar o no el marco)	Crea una figura
subplot(numFilas,numColumnas,numGrafica)	Crea varias gráficas en la misma ventana
plot(abcisas, ordenadas,colorytipo,anchoLinea,marcador)	Características de la gráfica

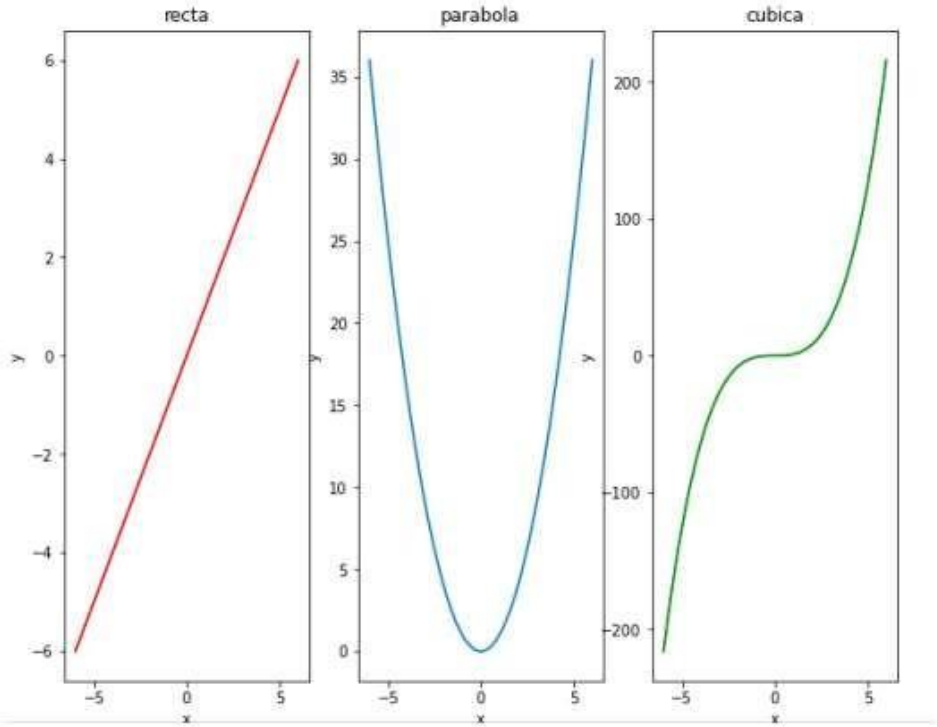
Matplotlib. Una única figura. Un axe

```
In [14]: import matplotlib.pyplot as plt
    ...: import numpy as np
    ...: #rango de las x
    ...: x = np.linspace(-6,6)
    ...: #figura y gráficas (un axis)
    ...: figura,graficas = plt.subplots()
    ...: graficas.plot(x, x, label='recta')
    ...: graficas.plot(x, x**2, label='parabola')
    ...: graficas.plot(x, x**3, label='cubica')
    ...:
    ...: #etiquetas y  título
    ...: graficas.set_xlabel('x')
    ...: graficas.set_title("Tres funciones")
Out[14]: Text(0.5, 1.0, 'Tres funciones')
```



Matplotlib. Una única figura, varios “axes”

```
In [15]: import matplotlib.pyplot as plt
    ...: import numpy as np
    ...:
    ...: #rango de las x
    ...: x = np.linspace(-6,6)
    ...: #dibuja una figura con tres axes (1 fila y tres columnas)
    ...: fig,ax=plt.subplots(1,3)
    ...: #fija  el anco y alto de la figura
    ...: fig.set_size_inches(10,8)
    ...:
    ...: #crea el gráfico de la primera columna (axe), en color rojo
    ...: ax[0].plot(x,x,label='recta', color="red" )
    ...: #crea el gráfico de la segunda columna (axe), en color por defecto
    ...: ax[1].plot(x,x**2,label='parabola')
    ...: #crea el gráfico de la tercera columna, en color verde
    ...: ax[2].plot(x,x**3,label='cubica', color="green")
    ...: #define las etiquetas de los ejes
    ...: for i in  range(3):
    ...:     ax[i].set_xlabel('x')
    ...:     ax[i].set_ylabel('y')
    ...: #define los nombre de cada gráfico
    ...: ax[0].set_title("recta")
    ...: ax[1].set_title("parabola")
    ...: ax[2].set_title("cubica")
Out[15]: Text(0.5, 1.0, 'cubica')
```



www.unir.net