

Ejercicio guiado. Unidad 6

- a. Generación de vectores y matrices usando numpy. Recordemos que los vectores y matrices de Numpy son homogéneas, por lo que todos sus componentes son del mismo tipo. Se pueden usar las funciones **array** y **matrix**.

```
import numpy as np

v = np.array([10, 20, 30, 40, 50])
m1 = np.matrix('10, 20, 40; 40, 50, 60; 70, 80, 90')
m2 = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
m3 = np.matrix([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12]).reshape(4, 3) # 4 filas, tres columnas
print(v)
print()
print(m1)
print()
print(m2)
print()
print(m1 + m2)
print()
print(m3)
[10 20 30 40 50]

[[10 20 40]
 [40 50 60]
 [70 80 90]]

[[1 2 3]
 [4 5 6]
 [7 8 9]]

[[11 22 43]
 [44 55 66]
 [77 88 99]]

[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

- b. Al igual que en Matlab, se dispone de dos funciones muy útiles: **arange**(inicio, final, paso) y **linspace**(inicio, final, número). Recordemos que usando la primera función el número final no aparecerá, pero si con la segunda.

```
x1 = np.arange(0, 10, 0.5)
x2 = np.linspace(0, 10, 25)
print(x1)
print()
print(x2)
```

[0. 0.5 1. 1.5 2. 2.5 3. 3.5 4. 4.5 5. 5.5 6. 6.5 7. 7.5 8. 8.5
9. 9.5]

```
[ 0.      0.41666667 0.83333333 1.25      1.66666667 2.08333333
 2.5      2.91666667 3.33333333 3.75      4.16666667 4.58333333
 5.       5.41666667 5.83333333 6.25      6.66666667 7.08333333
 7.5      7.91666667 8.33333333 8.75      9.16666667 9.58333333
10.      ]
```

- c. Dibujar un polinomio de tercer grado, de coeficientes 1, -6, -5 y 6, en un dominio de -4 a 8 tomando 50 puntos.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-4, 8, 50)
y1 = x ** 3 - 6 * x ** 2 - 5 * x + 6
y2 = 0 * x
plt.plot(x, y1, 'b-', x, y2, 'g-')
plt.title('Polinomio de tercer grado')
plt.xlabel('Abcisas')
plt.ylabel('Ordenadas')
plt.show()
```

