

## Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

### Tutorial Problema 7 de la Guía N° 4

**Problema 7:** Considere los vectores en  $\mathbb{R}^3$ :  $\vec{v}_1 = (1, 1, 1)$  y  $\vec{v}_2 = (0, 2, 3)$ . Calcule el producto interno, el producto vectorial y el producto elemento a elemento entre ellos.

---

#### Tutorial:

- Guarde en el archivo p7.py las siguientes instrucciones:

```
1 import numpy as np
2
3 print()
4 va = [1,1,1]
5 vb = [0,2,3]
6
7 va_ar = np.array([1,1,1])
8 vb_ar = np.array([0,2,3])
9
10 print(' va =', va)
11 print(' vb =', vb)
12 print(' va_ar =', va_ar)
13 print(' vb_ar =', vb_ar)
14
15
16 # tipos de objetos -----
17 print()
18 print(' tipos de objetos:')
19 print(' type(va) =', type(va))
20 print(' type(va_ar) =', type(va_ar))
21
22
23 # productos escalares -----
24 print()
25 print(' productos escalares:')
26 prodescalar = np.dot(va, vb)
27 print(' prodescalar = np.dot(va, vb) =', prodescalar)
28
29 prodescalar_ar = np.dot(va_ar, vb_ar)
30 print(' prodescalar_ar = np.dot(va_ar, vb_ar) =', prodescalar_ar)
31
32 print(' type(prodescalar) =', type(prodescalar))
33 print(' type(prodescalar_ar) =', type(prodescalar_ar))
34
35 print()
36
37 # productos vectoriales -----
```

```

38 print(' productos vectoriales:')
39
40 crossprod = np.cross(va,vb)
41 print('          crossprod = np.cross(va,vb) =',crossprod)
42
43 crossprod_ar = np.cross(va_ar,vb_ar)
44 print('crossprod_ar = np.cross(va_ar,vb_ar) =',crossprod_ar)
45
46
47 print('    type(crossprod) =',type(crossprod))
48 print('type(crossprod_ar) =',type(crossprod_ar))
49
50 print()
51
52 # producto por escalares -----
53 print(' producto por escalares:')
54
55 print('          5*va =',5*va)
56 print('          5*va_ar =',5*va_ar)
57 print()
58
59 # producto a lo bruto (no geométrico) -----
60 print(' producto directo (no geométrico):')
61
62 print(' va_ar*vb_ar =',va_ar*vb_ar)
63 print()

```

- Desde la terminal ejecute:

```
python3 p7.py
```

e interprete el resultado.

Alternativamente ejecute:

```
python3
```

y vaya agregando uno a uno los bloques del programa.

Obtendrá algo como esto:

```

1     va = [1, 1, 1]
2     vb = [0, 2, 3]
3 va_ar = [1 1 1]
4 vb_ar = [0 2 3]
5
6 tipos de objetos:
7     type(va) = <class 'list'>
8     type(va_ar) = <class 'numpy.ndarray'>
9
10 productos escalares:
11         prodescalar = np.dot(va,vb) = 5
12 prodescalar_ar = np.dot(va_ar,vb_ar) = 5
13     type(prodescalar) = <class 'numpy.int64'>
14 type(prodescalar_ar) = <class 'numpy.int64'>
15
16 productos vectoriales:
17         crossprod = np.cross(va,vb) = [ 1 -3  2]
18 crossprod_ar = np.cross(va_ar,vb_ar) = [ 1 -3  2]

```

```
19     type(crossprod) = <class 'numpy.ndarray'>
20 type(crossprod_ar) = <class 'numpy.ndarray'>
21
22 producto por escalares:
23     5*va = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
24 5*va_ar = [5 5 5]
25
26 producto directo (no geométrico):
27 va_ar*vb_ar = [0 2 3]
```