

# Soluciones a los ejercicios de Condicionales y bucles en Python (con NumPy)

## Ejercicio 1 (for): Mostrar los elementos de un vector

Dado el vector  $v = (1, 2, 3, 4, 5)$ , escribe un programa que muestre por pantalla todos sus elementos usando un bucle `for`.

```
import numpy as np

v = np.array([1, 2, 3, 4, 5])

for i in range(len(v)):
    print(v[i])
```

## Ejercicio 2 (for): Contar elementos positivos

Dado el vector  $v = (-2, 3, 0, 5, -1)$ , cuenta cuántos elementos son positivos.

```
import numpy as np

v = np.array([-2, 3, 0, 5, -1])

contador = 0

for i in range(len(v)):
    if v[i] > 0:
        contador = contador + 1

print("Cantidad de positivos:", contador)
```

### Ejercicio 3 (for): Producto de todas las entradas

Dado el vector  $v = (1, 2, 3, -4, 5)$ , calcula el producto de todas sus entradas usando un bucle `for`.

```
import numpy as np

v = np.array([1, 2, 3, -4, 5])

producto = 1

for i in range(len(v)):
    producto = producto * v[i]

print(producto)
```

### Ejercicio 4 (while): Primer elemento negativo

Para el vector  $v = (1, 2, 3, -4, 5)$ , busca el primer elemento negativo usando un bucle `while`.

```
import numpy as np

v = np.array([1, 2, 3, -4, 5])

i = 0

while i < len(v):
    if v[i] < 0:
        print("Primer negativo:", v[i])
        break
    i = i + 1
```

### Ejercicio 5 (for): Suma de todos los elementos de una matriz

Dada la matriz

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix},$$

calcula la suma de todos sus elementos usando bucles `for`.

```

import numpy as np

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

suma = 0

for i in range(2):
    for j in range(2):
        suma = suma + A[i, j]

print("Suma total:", suma)

```

## Ejercicio 6 (while): Primera columna con todos los elementos distintos de cero

Dada la matriz

$$B = \begin{pmatrix} 3 & \frac{2}{5} & 0 \\ 0 & -1 & 4 \\ \pi & 0 & 1 \end{pmatrix},$$

buscamos la primera columna cuyos elementos sean todos distintos de cero.

```

import numpy as np

B = np.array([
    [3, 2/5, 0],
    [0, -1, 4],
    [3.14, 0, 1]
])

j = 0

while j < 3:
    i = 0
    todos_distintos = True

    while i < 3:
        if B[i, j] == 0:
            todos_distintos = False
        i = i + 1

```

```

if todos_distintos:
    print("Primera columna válida:", j)
    break

j = j + 1

```

## Ejercicio 7 (for): Suma de las columnas

Calculamos la suma de cada columna de la matriz anterior.

```

import numpy as np

B = np.array([
    [3, 2/5, 0],
    [0, -1, 4],
    [3.14, 0, 1]
])

for j in range(3):
    suma = 0
    for i in range(3):
        suma = suma + B[i, j]
    print("Suma columna", j, "=", suma)

```

## Ejercicio 8: Método de Jacobi (versión muy básica)

Implementamos el método de Jacobi de la forma más simple posible.

```

import numpy as np

def jacobi(N, A, b, x, d):

    n = len(x)

    for k in range(N):

        x_nuevo = np.zeros(n)

```

```

for i in range(n):
    suma = 0
    for j in range(n):
        if j != i:
            suma = suma + A[i, j] * x[j]

    x_nuevo[i] = (b[i] - suma) / A[i, i]

diferencia = 0
for i in range(n):
    diferencia = diferencia + (x_nuevo[i] - x[i])**2

if diferencia < d*d:
    break

x = x_nuevo

return x

```