

TALLER PRACTICO NUMPY EN ACANCONDA

Juan David Echeverria García

**Profundización Disciplinar Ciencia De Datos
Séptimo Semestre**

**Universidad De Cundinamarca
Septiembre, 2025**

Ejercicio 1:

- Crea un arreglo de una dirección con los salarios de 5 empleados.
- Crea una matriz 2D con las ventas de 3 productos en 4 meses.
- Crea un arreglo 3D con temperaturas registradas en 2 ciudades, 3 días Y 4 mediciones por cada día.

▼ Ejercicio 1 ¶

```
[4]: salarios = np.array([1500000,2000000,1750000,2100000,1600000])
ventas = np.array([[10,12,11,10],[40,8,30,12],[9,11,5,10]])
temp = np.arange(24).reshape(2,3,4)
print("Salario de cada empleado:\n", salarios)
print("Ventas de 2 productos en 4 meses:\n", ventas)
print("Temperatura de 2 ciudades, 3 días y registro de 4 veces al día:\n",temp)

Salario de cada empleado:
[1500000 2000000 1750000 2100000 1600000]
Ventas de 2 productos en 4 meses:
[[10 12 11 10]
 [40  8 30 12]
 [ 9 11  5 10]]
Temperatura de 2 ciudades, 3 días y registro de 4 veces al día:
[[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]

 [[12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]]
```

Ejercicio 2:

- Usa las funciones shape, ndim y size para conocer las diferencias de los arreglos creados.

Ejercicio 2

```
[5]: tam_salarios = salarios.shape
tam_ventas = ventas.shape
tam_temp = temp.shape
dim_salarios = salarios.ndim
dim_ventas = ventas.ndim
dim_temp = temp.ndim
cant_salarios = salarios.size
cant_ventas = ventas.size
cant_temp = temp.size
print("Tamaño de salarios: ", tam_salarios)
print("Tamaño de ventas: ", tam_ventas)
print("Tamaño de temperaturas: ", tam_temp)
print("Dimensiones de salarios: ", dim_salarios)
print("Dimensiones de ventas: ", dim_ventas)
print("Dimensiones de temperaturas: ", dim_temp)
print("Cantidad elementos de salarios: ", cant_salarios)
print("Cantidad elementos de ventas: ", cant_ventas)
print("Cantidad elementos de temperaturas: ", cant_temp)

Tamaño de salarios: (5,)
Tamaño de ventas: (3, 4)
Tamaño de temperaturas: (2, 3, 4)
Dimensiones de salarios: 1
Dimensiones de ventas: 2
Dimensiones de temperaturas: 3
Cantidad elementos de salarios: 5
Cantidad elementos de ventas: 12
Cantidad elementos de temperaturas: 24
```

Ejercicio 3:

- Crea una base de datos aleatorios (inventario inicial).
- Buque la posición de un elemento en una matriz de ventas que sea mayor i que un valor determinado.
- Ordena los salarios.
- Eleva al cuadrado (power) las ventas.

Ejercicio 3

```
[6]: inventario = np.arange(12).reshape(3,4)
condicion = np.array(ventas>9)
orden_salarios = np.sort(salarios)
potencia = np.power(ventas,2)
print("Inventario:\n",inventario)
print("Mayores a 9:\n", condicion)
print("Salarios ordenados: ", orden_salarios)
print("Ventas al cuadrado:\n", potencia)

Inventario:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
Mayores a 9:
[[ True True True True]
 [ True False True True]
 [False True False True]]
Salarios ordenados: [1500000 1600000 1750000 2000000 2100000]
Ventas al cuadrado:
[[ 100  144  121  100]
 [1600   64   900  144]
 [  81  121   25  100]]
```

Ejercicio 4:

- Compara las ventas con un umbral de 30.
- Calcula la temperatura máxima y mínima.
- Concatena los salarios de dos sucursales.

Ejercicio 4

```
[7]: comparacion = np.array(ventas>=30)
temp_max = np.max(temp)
temp_min = np.min(temp)
suc1 = np.array([3000000,2800000,1500000,2300000,2400000])
suc2 = np.array([3540000,2400000,1250000,2600000,2600000])
sal_nuevos = salarios+suc1+suc2
print("Ventas mayores o iguales 30:\n",comparacion)
print(f"la temperatura maxima es: {temp_max} y la minima es {temp_min}")
print("Nuevos salarios: ", sal_nuevos)
```

```
Ventas mayores o iguales 30:
[[False False False False]
 [ True False  True False]
 [False False False False]]
la temperatura maxima es: 23 y la minima es 0
Nuevos salarios: [8040000 7200000 4500000 7000000 6600000]
```

Ejercicio 5:

- Realice operaciones múltiples (matriz de ventas + inventario), resta -5, multiplicar x2 y dividir entre 2.

▼ Ejercicio 5

```
[17]: suma = np.add(ventas,inventario)
      resta = np.subtract(ventas,5)
      mult = np.multiply(ventas,2)
      div = np.divide(ventas,2)
      print("Suma de ventas + Inventario:\n",suma)
      print("Resta de inventario -5:\n",resta)
      print("Multiplicar ventas x2:\n",mult)
      print("Dividir ventas /2:\n",div)
```

Suma de ventas + Inventario:

```
[[10 13 13 13]
 [44 13 36 19]
 [17 20 15 21]]
```

Resta de inventario -5:

```
[[ 5  7  6  5]
 [35  3 25  7]
 [ 4  6  0  5]]
```

Multiplicar ventas x2:

```
[[20 24 22 20]
 [80 16 60 24]
 [18 22 10 20]]
```

Dividir ventas /2:

```
[[ 5.  6.  5.5  5. ]
 [20.  4. 15.  6. ]
 [ 4.5  5.5  2.5  5. ]]
```

Ejercicio 6 (mini-proyecto):

- Una empresa tiene dos sucursales con datos de ventas de 3 productos en 4 meses.
- Calcule ventas totales, mes con mayor y menor tendido, comparando las dos sucursales.
- Concatena las matrices en una sola de 6x4.

Ejercicio 6 (mini-proyecto)

```
[36]: sucursal1 = np.array([[30,22,31,32],[40,28,35,22],[29,31,25,37]])
      sucursal2 = np.array([[25,32,21,31],[30,25,25,28],[23,33,35,34]])
      ventas_sucursales = np.add(sucursal1,sucursal2)
      ventas_totales = np.sum(ventas_sucursales)
      mes_suc1 = sucursal1.sum(axis=0)
      mes_suc2 = sucursal2.sum(axis=0)
      suc_unidas = np.concatenate((sucursal1,sucursal2))
      print("Sucursal 1: \n", sucursal1)
      print("Sucursal 2: \n", sucursal2)
      print("Ventas Totales de los 4 meses en ambas sucursales: ",ventas_totales)
      print(f"Meses con mayor venta\nSucursal 1: {np.max(mes_suc1)} - Sucursal 2: {np.max(mes_suc2)}")
      print(f"Meses con menor venta\nSucursal 1: {np.min(mes_suc1)} - Sucursal 2: {np.min(mes_suc2)}")
      print(f"Sucursales concatenadas {suc_unidas.shape}:\n",suc_unidas)
```

```
Sucursal 1:
[[30 22 31 32]
 [40 28 35 22]
 [29 31 25 37]]
Sucursal 2:
[[25 32 21 31]
 [30 25 25 28]
 [23 33 35 34]]
Ventas Totales de los 4 meses en ambas sucursales: 704
Meses con mayor venta
Sucursal 1: 99 - Sucursal 2: 93
Meses con menor venta
Sucursal 1: 81 - Sucursal 2: 78
Sucursales concatenadas (6, 4):
[[30 22 31 32]
 [40 28 35 22]
 [29 31 25 37]
 [25 32 21 31]
 [30 25 25 28]
 [23 33 35 34]]
```