



## ¿QUE SON LOS ARREGLOS -ARRAY-?

Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.

Scalar

24

Vector

$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$

row

or  
column  $\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$

Matrix

$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$

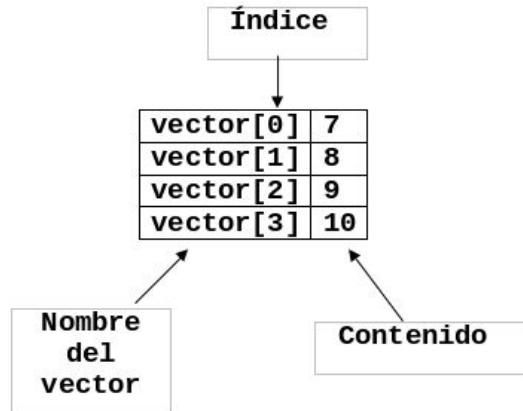
row(s) x column(s)





## ¿QUÉ SON LOS VECTORES?

Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.



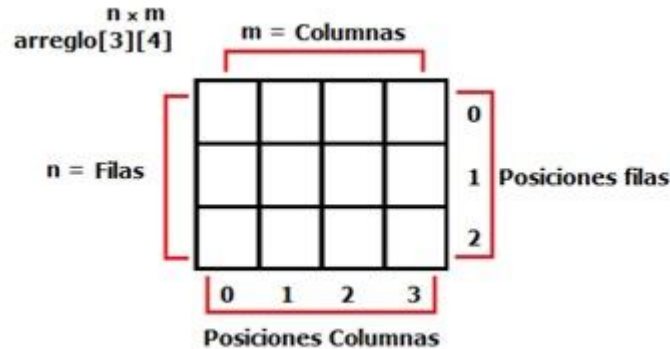
```
1  Proceso sin_titulo
2      Dimension datos(3);
3  FinProceso
4
```





## ¿QUÉ SON LAS MATRICES?

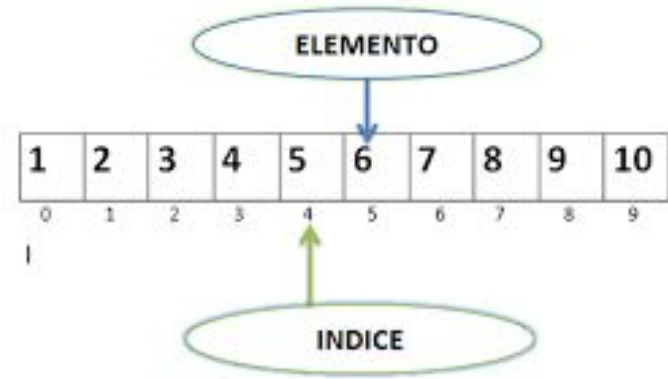
Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.





## ¿CÓMO CARGAR VECTORES?

```
1  Algoritmo carga_vector
2
3      Dimension vector(3)
4      Definir i Como Entero
5      i ← 0
6      //los vectores en PseInt comienzan con 1
7      Para i ← 1 Hasta 3
8          Escribir "Ingrese un número para la posición: " i
9          Leer vector(i)
10
11      FinPara
12
13  FinAlgoritmo
```

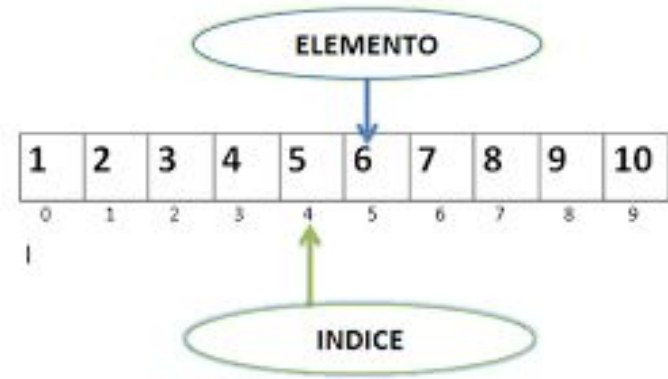




## ¿CÓMO RECORRER VECTORES?

```
Algoritmo recorrido_vector
    //Carga de vector
    Dimension vector(3)
    Definir i Como Entero
    vector(1) ← 35
    vector(2) ← 12
    vector(3) ← 58
    i ← 0
    //Recorrido de vector
    Para i ← 1 Hasta 3 Con Paso 1
        Mostrar vector(i)
    FinPara
```

FinAlgoritmo





## EJERCICIOS CON VECTORES

1. Carga un vector de 5 elementos con números enteros al azar y luego imprime sus valores en pantalla.
2. Carga dos vectores de 5 elementos cada uno con números enteros al azar y luego realiza la suma componente a componente de ambos vectores, almacenando el resultado en un tercer vector. Imprime el vector resultado.
3. Carga un vector de 10 elementos con números reales y calcula su promedio. Imprime el promedio en pantalla.



## ¿CÓMO CARGAR MATRICES?

```
Algoritmo Carga_matriz
2
3   Dimension  matriz(2,3)
4
5   Definir fila, columna Como Entero
6
7   matriz(0,0)←1
8   matriz(0,1)←5
9   matriz(0,2)←8
10  matriz(1,0)←7
11  matriz(1,1)←3
12  matriz(1,2)←2
13
14  Para fila ← 0 Hasta 1 Con Paso 1
15    Para columna←0 Hasta 2 Con Paso 1
16      Mostrar  matriz(fila, columna), " " Sin Saltar
17
18    FinPara
19
20    Escribir ""
21
22  FinPara
23
24 FinAlgoritmo
```

		Columnas			
		0	1	2	3
Filas	0	2	4	5	8
	1	6	3	1	9



## ¿CÓMO RECORRER MATRICES?

Algoritmo Carga\_matriz

Dimension matriz(6,6)

Definir fila, columna Como Entero

Para fila  $\leftarrow$  0 Hasta 5 Con Paso 1

Para columna  $\leftarrow$  0 Hasta 5 Con Paso 1

Si fila == columna Entonces

matriz(fila, columna)  $\leftarrow$  1

FinSi

FinPara

FinPara

Para fila  $\leftarrow$  0 Hasta 5 Con Paso 1

Para columna  $\leftarrow$  0 Hasta 5 Con Paso 1

Mostrar matriz(fila, columna), " " Sin Saltar

FinPara

Escribir " "

FinPara

FinAlgoritmo

		Columnas			
		0	1	2	3
Filas	0	2	4	5	8
	1	6	3	1	9







## EJERCICIOS CON MATRICES

1. Crea una matriz de  $4 \times 4$  y carga sus valores de manera aleatoria. Pide al usuario que ingrese un número y muestra cuántas veces aparece ese número en la matriz.
2. Crea una matriz cuadrada de  $4 \times 4$  y carga sus valores de manera aleatoria. Calcula la suma de los elementos en la diagonal principal de la matriz e imprime el resultado.
3. Crea una matriz de  $3 \times 3$  y carga sus valores de manera aleatoria. Pide al usuario que ingrese un número y busca si el número se encuentra en la matriz. Si el número se encuentra en la matriz, indica su posición en la matriz.

