

LENGUAJE DE PROGRAMACIÓN C

ESTRUCTURAS DE DATOS EN C/C++

ARREGLOS (VECTORES Y MATRICES)

Las estructuras de datos están subdivididas por estáticas (espacio fijo establecido en memoria) y dinámicas (sin restricciones y limitaciones en el espacio usado en memoria).

Estructuras de datos estáticas

- *Arrays* (vectores y matrices)
- Cadenas
- Registros
- Ficheros

Estructuras de datos dinámicas

- Listas (pilas y colas)
- Listas enlazadas
- Árboles
- Grafos

La diferencia entre cada estructura es la forma de cómo se almacena y manipula el conjunto de datos, permitiendo así su eficiencia en el resultado de una operación sobre dichos datos.

ARREGLOS (VECTORES Y MATRICES)

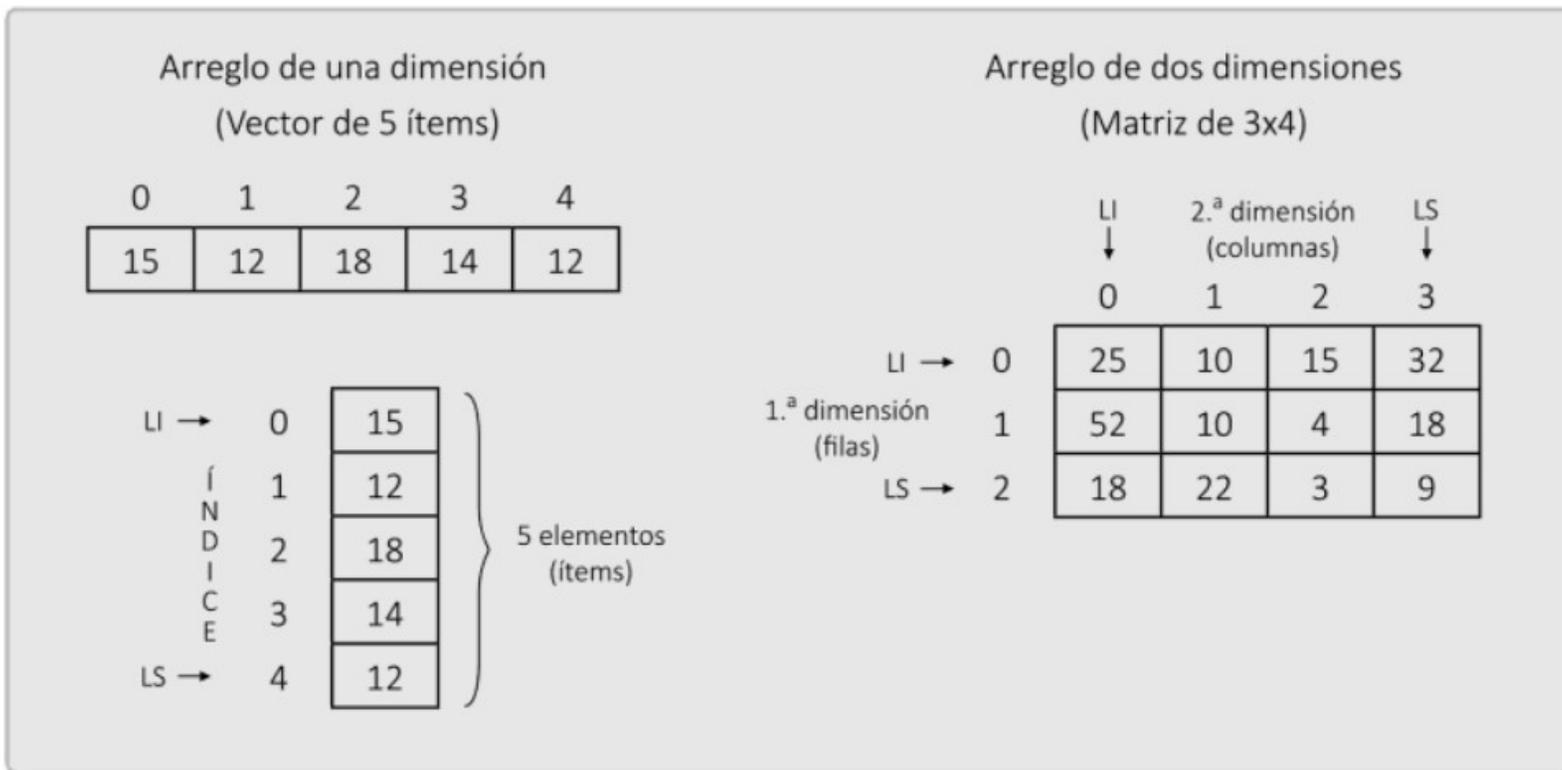
Arrays (arreglos)

Es un conjunto finito (tamaño fijo) y ordenado (usa un índice) de datos homogéneos (datos del mismo tipo). Los arreglos pueden ser de una dimensión (vector), dos dimensiones (matriz) y n dimensiones (multidimensional).

En todos los lenguajes de programación, los **arrays** usan un índice numérico para cada elemento que contiene, los cuales por lo general inician con el índice 0, llamado «Limite Inferior» (LI); y el último elemento tendrá el índice llamado «Limite Superior» (LS), que en sí es la cantidad de elementos del **array** menos 1.

Vector					Matriz				
0	1	2	3	4		0	1	2	3
15	12	18	14	12	0	25	10	15	32
					1	52	10	4	18
					2	18	22	3	9

ARREGLOS (VECTORES Y MATRICES)



ARREGLOS (VECTORES Y MATRICES)

Operaciones con *arrays*

Las operaciones son el procesamiento y el tratamiento individual de los elementos del *array*, los cuales son los siguientes:

- Asignación
- Lectura / Escritura
- Recorrido
- Actualización (insertar, borrar, modificar)
- Ordenación
- Búsqueda

ARREGLOS (VECTORES Y MATRICES)

Creación de *arrays*

Para la creación de un *array* se requiere conocer el nombre, las dimensiones, el tamaño de elementos y el tipo de dato.

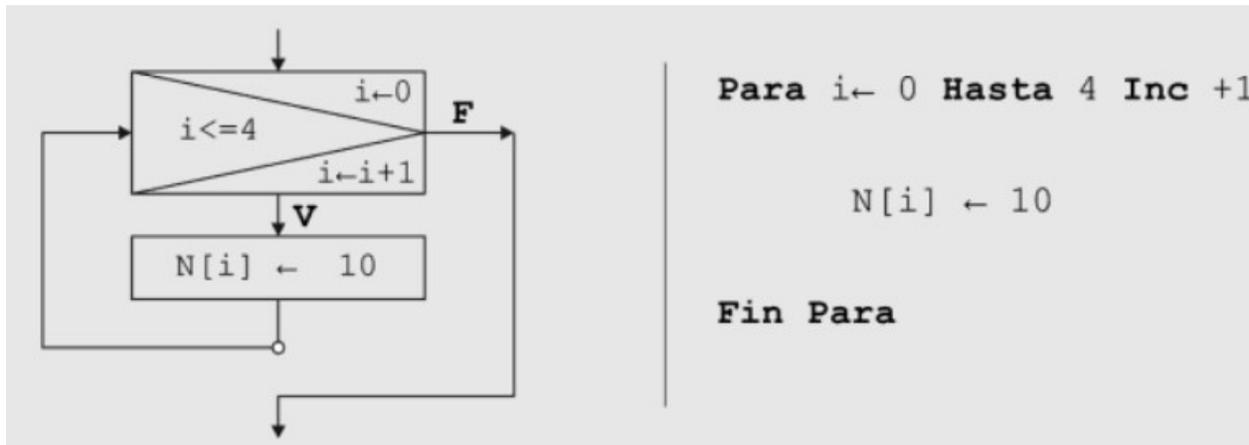
```
//Array de una dimensión (Vector)
// 5 elementos LI = 0 y LS = 4
int N[5];
```

```
//Array de dos dimensiones (Matriz)
// 3X4 elementos
// 1era Dim. LI = 0 y LS = 2
// 2da Dim. LI = 0 y LS = 3
int N[3][4];
```

ARREGLOS (VECTORES Y MATRICES)

Recorrido por los elementos del *array*

Para realizar un recorrido por cada elemento del *array* utilizaremos la estructura repetitiva «Para» (For).
En el siguiente diagrama se tiene el **vector** N de 5 elementos, y se asigna el valor 10 a cada elemento.

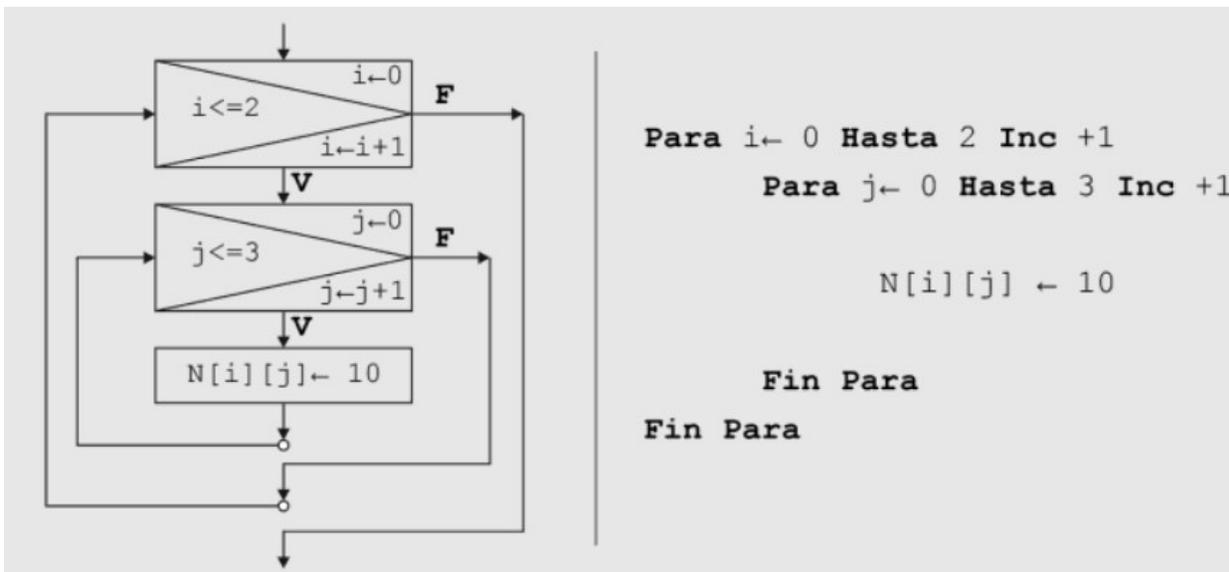


```
Para i ← 0 Hasta 4 Inc +1  
    N[i] ← 10  
Fin Para
```

```
for (i=0; i<=4; i++) {  
    N[i] = 10;  
}
```

ARREGLOS (VECTORES Y MATRICES)

En el siguiente diagrama se tiene la **matriz** N de 3x4 elementos y se asigna el valor 10 a cada elemento.



```
for (i=0;i<=2;i++) {
  for (j=0;j<=3;j++) {
    N[i][j] = 10;
  }
}
```