

USO DE ARREGLOS EN ESTRUCTURAS DE DATOS

- **Matrices (arreglos de 2D o más dimensiones)** → como una tabla.
- **Listas enlazadas** → parecidas a arrays pero con nodos conectados.
- **Pilas (Stack)** → como una pila de platos (LIFO: el último que entra es el primero que sale).
- **Colas (Queue)** → como una fila en el banco (FIFO: el primero en entrar es el primero en salir).
- **Árboles y grafos** → estructuras más avanzadas para representar jerarquías o conexiones.
- **Algoritmos de búsqueda y ordenamiento** → cómo organizar y encontrar datos rápido.

Lenguaje	Declaración vacía/fija	Inicialización con valores	Acceso por índice	Tamaño
Python	<code>[]</code> o <code>[0]*n</code>	<code>[10,20,30]</code>	<code>arr[0]</code>	Dinámico
Java	<code>new int[5]</code>	<code>{10,20,30}</code>	<code>arr[0]</code>	Fijo
C#	<code>new int[5]</code>	<code>{10,20,30}</code>	<code>arr[0]</code>	Fijo
C++	<code>int arr[5];</code>	<code>{10,20,30}</code>	<code>arr[0]</code>	Fijo (o <code>vector</code>)
JS	<code>[]</code>	<code>[10,20,30]</code>	<code>arr[0]</code>	Dinámico

PYTHON

```
arreglo = []          # Lista vacía
arreglo = [10, 20, 30] # Lista con valores iniciales

# Asignar un valor mediante índice
arreglo[0] = 99 # ahora el primer elemento vale 99

valor = arreglo[1] # guarda 20

# Inicializar con valores por defecto
arreglo = [0] * 5 # crea [0, 0, 0, 0, 0]

# Leer un valor (ejemplo en bucle)
for i in range(len(arreglo)):
    print(f"Índice {i} -> {arreglo[i]}")
```

C++

```
#include <iostream>
using namespace std;
int main() {
    // Declaración (tamaño fijo)
    int arreglo[5]; // valores indefinidos

    // Inicialización
    int arreglo2[5] = {10, 20, 30, 40, 50};

    // Asignar valor
    arreglo[0] = 99;

    // Recuperar valor
    int valor = arreglo2[2]; // 30

    // Recorrer arreglo
    for(int i = 0; i < 5; i++) {
        cout << "Índice " << i << " -> " << arreglo2[i] << endl;
    }

    return 0;
}
```

C#

```
// Declaración
int[] arreglo = new int[5]; // inicializado en 0

// Inicialización con valores
int[] arreglo2 = {10, 20, 30, 40, 50};

// Asignar valor por índice
arreglo[1] = 77;

// Recuperar un valor
int valor = arreglo2[3]; // 40

// Recorrer arreglo
for(int i = 0; i < arreglo2.Length; i++) {
    Console.WriteLine("Índice " + i + " -> " + arreglo2[i]);
}
```

JAVA

```
int[] arreglo = new int[5]; // arreglo de 5 enteros inicializados en 0

// Inicialización con valores
int[] arreglo2 = {10, 20, 30, 40, 50};

// Asignar valor por índice
arreglo[0] = 99;

// Recuperar un valor
int valor = arreglo2[2]; // 30

// Recorrer un arreglo
for(int i = 0; i < arreglo2.length; i++) {
    System.out.println("Índice " + i + " -> " + arreglo2[i]);
}
```

JAVASCRIPT

```
// Declaración
let arreglo = [];

// Inicialización
let arreglo2 = [10, 20, 30, 40, 50];

// Asignar valor
arreglo2[0] = 99;

// Recuperar valor
let valor = arreglo2[2]; // 30

// Recorrer arreglo
for (let i = 0; i < arreglo2.length; i++) {
    console.log("Índice " + i + " -> " + arreglo2[i]);
}
```

Lenguaje	Declarar arreglo vacío/fijo	Inicializar con valores	Asignar valor con índice	Acceder a valor con índice	Recorrer arreglo
Python	<code>arr = [0]*3</code> (3 elementos en 0)	<code>arr = [5,10,15]</code>	<code>arr[0] = 99</code>	<code>valor = arr[1]</code>	<code>for x in arr: print(x)</code> o <code>for i in range(len(arr)): print(arr[i])</code>
Java	<code>int[] arr = new int[3];</code>	<code>int[] arr = {5,10,15};</code>	<code>arr[0] = 99;</code>	<code>int v = arr[1];</code>	<code>for(int i=0; i<arr.length; i++){System.out.println(arr[i]);}</code>
C#	<code>int[] arr = new int[3];</code>	<code>int[] arr = {5,10,15};</code>	<code>arr[0] = 99;</code>	<code>int v = arr[1];</code>	<code>for(int i=0; i<arr.Length; i++){Console.WriteLine(arr[i]);}</code>
C++	<code>int arr[3];</code>	<code>int arr[3] = {5,10,15};</code>	<code>arr[0] = 99;</code>	<code>int v = arr[1];</code>	<code>for(int i=0; i<3; i++){cout << arr[i];}</code>
JavaScript	<code>let arr = [];</code> o <code>let arr = [0,0,0];</code>	<code>let arr = [5,10,15];</code>	<code>arr[0] = 99;</code>	<code>let v = arr[1];</code>	<code>for(let i=0; i<arr.length; i++){console.log(arr[i]);}</code>