

# Resumen de guía 5

## Parte I - Arreglos

# Arreglos

Un **arreglo** es un contenedor de objetos en el que **se almacenan un número fijo de valores del mismo tipo**. El tamaño del arreglo se establece en el momento de su creación y permanece constante durante toda su vida útil, lo que significa que no puede cambiar dinámicamente.

Cada elemento en un arreglo tiene una posición única llamada **índice**, que **se utiliza para acceder y manipular los valores almacenados**.

La ventaja de utilizar arreglos es que proporcionan un **acceso eficiente a los elementos almacenados**, ya que los elementos están dispuestos en una ubicación contigua a la memoria. Esto permite un acceso rápido y directo a cualquier elemento utilizando su índice.


Sin embargo, la principal limitación de los arreglos es su tamaño fijo.

# Declaración de arreglos

Tipo[] nombre;  declaración del arreglo

nombre=new Tipo[10];  asignación de memoria

Podemos recorrer el arreglo usando un for each o un for tradicional

```
for (Tipo elem: nombre){  recorrer el arreglo  
    print elem;  
}
```

# Ejemplo arreglos

0	1	2	3	4

arreglo

```
public class ArregloEjemplo {
    public static void main(String[] args) {
        String[] arreglo = new String[5];
        llenarArreglo(arreglo);
        mostrarArreglo(arreglo);
        buscarElemento(arreglo);
    }

    public static void llenarArreglo(String[] arreglo) {
        Scanner scanner = new Scanner(System.in);
        System.out.println(x: "Llenando el arreglo...");
        for (int i = 0; i < arreglo.length; i++) {
            System.out.print("Ingrese el elemento en la posición " + i + ": ");
            arreglo[i] = scanner.nextLine();
        }
    }

    public static void mostrarArreglo(String[] arreglo) {
        System.out.println(x: "\nMostrando el arreglo...");
        for (String i: arreglo) {
            System.out.print( "[" + i + "]" );
        }
        System.out.println(x: "");
    }

    public static void buscarElemento(String[] arreglo) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "\nIngrese el elemento a buscar: ");
        String elementoBuscado = scanner.nextLine();
        int posicion = -1;
        for (int i = 0; i < arreglo.length; i++) {
            if (arreglo[i].equals(elementoBuscado)) {
                posicion = i;
                System.out.println("El elemento '" + elementoBuscado + "' se encuentra en la posición " + posicion);
                break;
            }
        }
        if (posicion == -1) {
            System.out.println("El elemento '" + elementoBuscado + "' no se encuentra en el arreglo.");
        }
    }
}
```

# Arreglos

**Arreglos como argumento:** Cuando se pasa un arreglo a un método, cualquier modificación realizada en los elementos del arreglo dentro del método se reflejará en el arreglo original, ya que ambos apuntan a la misma ubicación de memoria.

```
public void listar(String[] arreglo){  
    for (String elem:arreglo){  
        System.out.println("Elemento: "+elem);  
    }  
}
```

Retornar un arreglo

```
public String[] devolverVector(){  
    String vector[]={ "uno", "dos", "tres" };  
    return vector;  
}
```

# Matrices

```
public class matrices {  
    public static void main(String[] args) {  
        int[][] matriz=new int[3][3];  
        llenarMatriz(matriz);  
        mostrarMatriz(matriz);  
    }  
  
    // Método para llenar la matriz con valores  
    public static void llenarMatriz(int[][] matriz) {  
        int contador = 1;  
        for (int i = 0; i < matriz.length; i++) {  
            for (int j = 0; j < matriz.length; j++) {  
                matriz[i][j] = contador;  
                contador++;  
            }  
        }  
    }  
  
    // Método para mostrar los elementos de la matriz  
    public static void mostrarMatriz(int[][] matriz) {  
        for (int i = 0; i < matriz.length; i++) {  
            for (int j = 0; j < matriz.length; j++) {  
                System.out.print(matriz[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

matriz

```
public class MatrizIrregular {  
    public static void main(String[] args) {  
        int[][] matrizIrregular = new int[3][];  
        matrizIrregular[0] = new int[2];  
        matrizIrregular[1] = new int[3];  
        matrizIrregular[2] = new int[4];  
  
        matrizIrregular[0][0] = 1;  
        matrizIrregular[0][1] = 2;  
  
        matrizIrregular[1][0] = 3;  
        matrizIrregular[1][1] = 4;  
        matrizIrregular[1][2] = 5;  
  
        matrizIrregular[2][0] = 6;  
        matrizIrregular[2][1] = 7;  
        matrizIrregular[2][2] = 8;  
        matrizIrregular[2][3] = 9;  
  
        for (int i = 0; i < matrizIrregular.length; i++) {  
            for (int j = 0; j < matrizIrregular[i].length; j++) {  
                System.out.print(matrizIrregular[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

0		→	1	2		
1		→	3	4	5	
2		→	6	7	8	9
			0	1	2	3

matrizIrregular