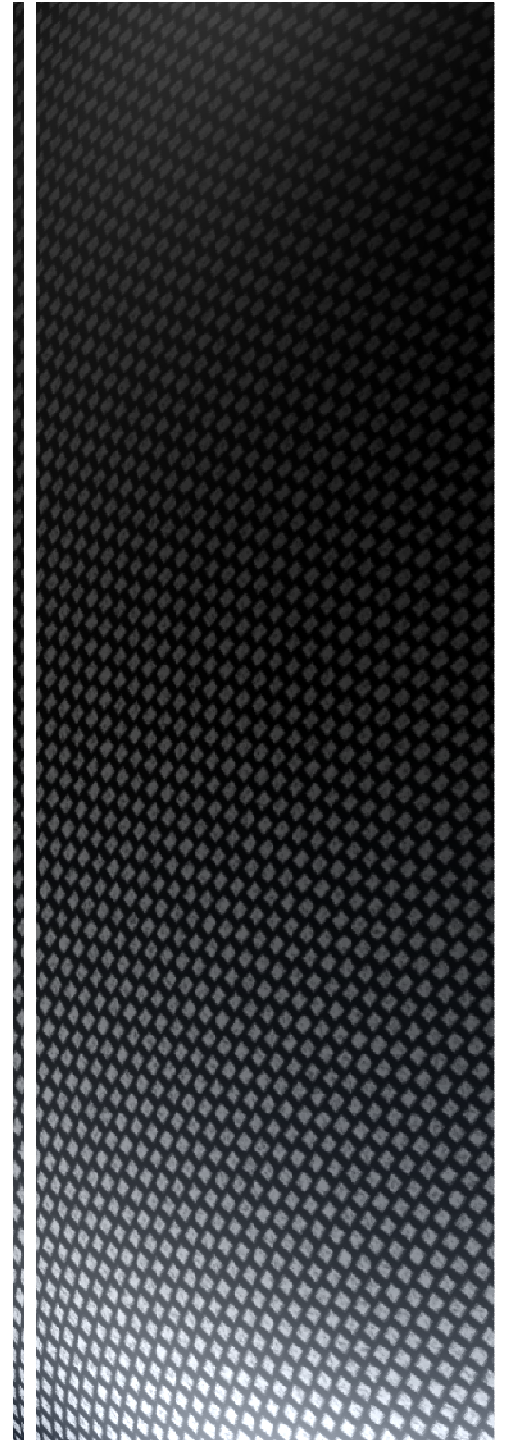


# Arrays

Unidad 6



# Contenidos

- Definición
- Arrays unidimensionales
- Arrays de varias dimensiones
- Ordenación y búsqueda en arrays
- Otras funciones útiles de arrays

# Arrays

## Definición

**Los arrays son estructuras de datos que permiten almacenar un conjunto de valores del mismo tipo.**

Pueden ser:

- Arrays unidimensionales (vectores)

valor	5	9	6	-4	0	7
índice	0	1	2	3	4	5

- Arrays multidimensionales (matrices)

		columnas			
		0	1	2	3
filas	0	8	4	2	-6
	1	-2	6	5	12
	2	3	1	0	7

# Arrays unidimensionales (vectores)

## Definición

Un **array** (o **vector**) es un conjunto finito de valores homogéneos (del mismo tipo) y ordenado, es decir, cada valor está identificado por una posición concreta del vector.

- El **índice** va de 0 a n-1 (**n**: cantidad total de elementos del array)
- **Declaración en java:**

```
int[] vector1;
```

- `[]`: identifican que el array es de una dimensión.

# Arrays unidimensionales (vectores)

## Inicialización

**Inicializar el array:** indicar cuantos elementos va a tener (se reserva el espacio necesario en memoria para albergar los elementos del vector).

```
int[] vector1;  
vector1 = new int[3];  
  
int[] vector2 = {7,42,10};
```

# Arrays unidimensionales (vectores)

Asignación de un nuevo valor en una posición

Para asignar un nuevo valor a una posición del array, por ejemplo al índice 2 sería:

```
vector1[2] = 12;  
System.out.println("Vector1[2] = " + vector1[2]);
```

Salida por pantalla:

```
Vector1[2] = 12
```

# Arrays unidimensionales (vectores)

## Recorrer un array

```
// Declaramos el primer array
int[] vector1;
vector1 = new int[3];

// Declaramos el segundo array
int[] vector2 = {7,42,10};

// Damos valores al primer array
vector1[0] = 4;
vector1[1] = 8;
vector1[2] = 12;

// Imprimimos el primer array
System.out.println("vector1:");
for(int i=0; i<3; i++){
    System.out.println(vector1[i]);
}

// Imprimimos el segundo array
System.out.println("\nvector2:");
for(int i=0; i<vector2.length; i++){
    System.out.println(vector2[i]);
}
```

```
vector1:
4
8
12

vector2:
7
42
10
```

# Arrays de varias dimensiones (matrices)

## Definición

Se trata de arrays que a su vez contienen otros arrays.

- El **índice** de cada dimensión va de 0 a n-1.
- **Declaración en java:**

```
int[][] matriz;
```

- `[][]`: identifican que el array es de dos dimensiones.



# Arrays de varias dimensiones (matrices)

## Inicialización

La **instanciación** y **acceso** es similar al caso de los arrays unidimensionales.

```
int[][] matriz1;  
matriz1 = new int[2][3];  
  
matriz1[1][2] = 34;  
  
int[][] matriz2 = {{3,2,0},{12,7,2}};
```

# Arrays de varias dimensiones (matrices)

Recorrer un array bidimensional

```
public class Matriz {  
    public static void main(String args[]) {  
        // Declaramos la matriz  
        int[][] matriz = {{3,2,0},{12,7,2}};  
  
        // Imprimimos la matriz  
        System.out.println("Matriz:");  
        for(int i=0; i<2; i++) // filas  
            for(int j=0; j<3; j++) // columnas  
                System.out.println(matriz[i][j]);  
    }  
}
```

Matriz:  
3  
2  
0  
12  
7  
2