

## Contenido

1. INTRODUCCIÓN A LAS TABLAS O ARRAYS UNIDIMENSIONALES.....	2
2. INTRODUCCIÓN A LOS ARRAYS BIDIMENSIONALES. ....	7

## 1. INTRODUCCIÓN A LAS TABLAS O ARRAYS UNIDIMENSIONALES.

Una **tabla** o **array** (que algunos autores traducen por "**arreglo**") es un conjunto de elementos, todos son del mismo tipo, todos tienen el mismo nombre y ocuparán un espacio contiguo en la memoria. Para acceder a cada elemento usamos un **índice**. Por ejemplo, definimos una tabla para almacenar 4 enteros, la podemos representar de esta manera:

Tabla <b>Numeros</b>	34	4	29	45
Nº de elemento o Índice	0	1	2	3

Podemos acceder a cada uno de los valores individuales indicando su nombre (en este caso la tabla se llama **Numeros**) y el número de elemento que nos interesa (también llamado **índice**). (En Java hemos de tener precaución porque se empieza a numerar desde 0).

En el ejemplo anterior tenemos 4 elementos que serían:

```
Numeros[0] tiene el valor 34
Numeros[1] tiene el valor 4
Numeros[2] tiene el valor 29
Numeros[3] tiene el valor 45
```

**Para declarar una tabla o array de dimensión N** (N es el número de elementos del array) **y cada elemento del tipo TipoDato** escribimos lo siguiente:

```
TipoDato NombreArray[] = new TipoDato[N];
TipoDato [] NombreArray = new TipoDato[N];
```

**Para dar valor a los elementos del array** tengo que saber a qué posición le asignaré el valor, para ello uso el índice:

```
NombreTabla[índice] = valor.
```

**Por ejemplo, para declarar una tabla o array de N elementos enteros** podemos escribir lo siguiente:

```
int Numeros[] = new int[N];
```

o bien:

```
int [] Numeros = new int[N];
```

**Ejemplo:**

```
int N = 4;
int Numeros[] = new int[N];
Numeros[0] = 34;
Numeros[1] = 4;
Numeros[2] = 29;
```

```
Numeros[3] = 45;
```

El siguiente ejemplo define una tabla de nombres de persona:

Tabla Nombres				
Nº de elemento o Indice	0	1	2	3
	"Ana"	"Juan"	"Andres"	"Maria"

```
Nombres[0] tiene el valor "Ana"
Nombres[1] tiene el valor "Juan"
Nombres[2] tiene el valor "Andres"
Nombres[3] tiene el valor "Maria"
```

Para declarar una tabla o array de N elementos cadena de caracteres y de nombre *Nombres* escribimos lo siguiente:

```
String [] Nombres = new String [N];
```

o bien:

```
String Nombres [] = new String [N];
```

Ejemplo:

```
int N = 4;
String [] Nombres = new String [N];
Nombres[0] = "Ana";
Nombres[1] = "Juan";
Nombres[2] = "Andres";
Nombres[3] = "Maria";
```

Para recorrer los elementos de un array usamos un bucle **for**, que comienza en el valor inicial del índice y finaliza en el valor final del índice:

Ejemplos:

Visualizo los datos de la de la tabla *Numeros* y obtengo la suma:

```
int suma = 0;
for(int i=0; i < N; i++) {
    System.out.printf("Valor elemento %d: %d\n",i, Numeros[i]);
    suma += Numeros[i];
}
System.out.println("La suma es: " + suma);
```

```
Valor elemento 0: 34
Valor elemento 1: 4
Valor elemento 2: 29
Valor elemento 3: 45
La suma es: 112
```

Visualizo los nombres de la tabla *Nombres*:

```
for(int i=0; i < N; i++) {
    System.out.printf("Nombre: %s\n",Nombres[i]);
}
```

```
Nombre: Ana
Nombre: Juan
Nombre: Andres
Nombre: Maria
```

DEFINIR UN ARRAY PAR A 10 ENTEROS. LLENAR EL ARRAY CON NUMEROS ALEATORIOS (1 A 100). Y DESPUES VISUALIZAR CONTENIDO.

**Inicializar arrays en la definición:**

```
int t[] = { 10, 20, 30, 40, 50 };

for (int i = 0; i < t.length; i++) {
    System.out.printf("Elemento %d: %d\n", i, t[i]);
}

String dias[] = { "Lunes", "Martes", "Miercoles", "Jueves",
                  "Viernes", "Sabado", "Domingo" };

for (int i = 0; i < dias.length; i++) {
    System.out.printf("Elemento %d: %s\n", i, dias[i]);
}
```

#### EJEMPLOS:

**Ejemplo 1)** Crea un array con un tamaño para 5 enteros. Asígnale los valores numéricos manualmente (los que tu quieras) y muéstralos por pantalla.

```
public class ejemplo1 {

    public static void main(String[] args) {
        int num[] = new int[5];
        // Asignamos valores
        num[0] = 10;
        num[1] = 10;
        num[2] = 20;
        num[3] = 30;
        num[4] = 40;

        // Recorremos el array y mostramos su contenido
        for (int i = 0; i < 5; i++) {
            System.out.printf("Valor elemento %d: %d\n", i, num[i]);
        }
    }

}

} //FIN
```

**Ejemplo 2)** En este caso llenaremos el array a partir de datos introducidos por teclado, usaremos un bucle para ello. Y mostramos el contenido con la clase **Arrays**.

```
import java.util.Scanner;
import java.util.Arrays;

public class ejemplo2 {
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        int num[] = new int[5];

        // Recorremos el array y mostramos su contenido
        for (int i = 0; i < 5; i++) {
            System.out.printf("Dame valor para el elemento %d: ", i);
            num[i] = sc.nextInt();
        }
        //Visualizamos contenido del array
        System.out.println(Arrays.toString(num));
    }
}

// FIN
```

**Ejemplo 3)** Llena un array de números aleatorios para 10 elementos. Muéstralos por pantalla. Calcula la suma, la media, máximo y posición, mínimo y posición.

```
import java.util.Arrays;
public class ejemplo3 {

    public static void main(String[] args) {

        int num[] = new int[10];
        // llenamos con numeros entre 1 y 100
        for (int i = 0; i < 10; i++) {
            num[i] = (int) (Math.random() * (100) + 1);
        }
        // Visualizamos contenido del array
        System.out.println(Arrays.toString(num));

        int max = -1, pmax=-1, min = 200, pmin=-1, suma=0;

        for (int i = 0; i < 10; i++) {
            if (num[i] > max) {
                max = num[i];
                pmax = i;
            }
            if (num[i] < min) {
                min = num[i];
                pmin = i;
            }
            suma += num[i];
        }

        float media = suma/10f;
        System.out.printf("Maximo: %d, posición: %d %n", max, pmax);
        System.out.printf("Minimo: %d, posición: %d %n", min, pmin);
        System.out.printf("Suma: %,5d, Media: %,6.2f %n", suma, media);
    }
}

// FIN
```

**Ejemplo 4)** Función que recibe un entero, N, y devuelve un array de N enteros entre 1 y 100 obtenidos aleatoriamente. Desde le método **main()** se llama a este método para llenar el array y después se muestra en pantalla.

```
import java.util.Arrays;

public class ejemplo4 {

    private static int[] dameArray(int N) {
        int[] num = new int[N];
        for (int i = 0; i < N; i++) {
            num[i] = (int) (Math.random() * (100) + 1);
        }
        return num;
    }

    public static void main(String[] args) {
        int N=10;
        int num[] = dameArray(N);
        System.out.println("Numero de elementos: " + num.length);
        System.out.println(Arrays.toString(num));

        int num2[] = dameArray(5);
        System.out.println("Numero de elementos: " + num2.length);
        System.out.println(Arrays.toString(num2));
    }
}

// FIN
```

**Ejemplo 5)** Método que recibe un array y lo llena con números aleatorios. Desde **main()** se llama a este método para llenar el array y después se muestra en pantalla.

```
import java.util.Arrays;
public class ejemplo5 {

    private static void llenaArray(int[] t) {
        for (int i = 0; i < t.length; i++) {
            t[i] = (int) (Math.random() * (100) + 1);
        }
    }

    public static void main(String[] args) {
        int N=10;
        int[] num = new int[N];
        System.out.println(Arrays.toString(num)); //ceros
        llenaArray(num);
        System.out.println(Arrays.toString(num));
    }
}

// FIN
```

### EJERCICIOS:

### CREAR Y LLENAR UN ARRAY DE 10 ELEMENTOS CON NUMEROS DEL 1 AL 100, en main() Y LUEGO HACER ESTOS MÉTODOS

- >> Realiza una función que reciba un array de enteros y devuelva el valor máximo.
- >> Realiza una función que reciba un array de enteros y devuelva la posición donde está el valor máximo.
- >> Realiza una función que reciba un array de enteros y un número entero y devuelva el número de veces que aparece dicho número en el array, si el número no aparece debe devolver -1.

>> Realiza una función que MUESTRE EL NÚMERO DE VECES QUE SE REPITE CADA ELEMENTO DEL ARRAY.

(ejercicioFunciones.java)

>> Introducir por teclado el nombre y la edad de **5 personas**. Esos datos se irán guardando en 2 arrays uno para los nombres y otro para las edades. En la posición 0 de cada array irá el nombre de una persona y su edad. En la posición 1 va el nombre de otra persona y su edad, y así sucesivamente ....

Una vez que has llenado el array, haz un método que reciba los arrays y una edad, y muestre el nombre de las personas que tienen esa edad. Si no hay ninguna se mostrará un mensaje indicándolo.

Leer una edad por teclado y llamar a la función para visualizar los nombres con esa edad.

(ejercicioFuncionesNombreEdad.java)

VER PÁGINAS 128, 129,130

>>REALIZAR UNA FUNCIÓN QUE **DEVUELVA UN ARRAY** DE NUMEROS ENTEROS DE LONGITUD **N** (PARÁMETRO QUE RECIBE LA FUNCIÓN) CON NUMEROS **PARES** DE 1 A 100 GENERADOS ALEATORIAMENTE. REALIZA UN PROGRAMA JAVA QUE HAGA USO DE LA FUNCION (main()). VISUALIZAR EL CONTENIDO DEL ARRAY DESPUES DE LA LLAMADA.(generaAleatorioPar.java)

>> REALIZA UNA FUNCION QUE **RECIBA UN ARRAY DE ENTEROS (llenar con números aleatorios el array) Y UN NÚMERO ENTERO (N)** Y DEVUELVA **OTRO ARRAY (el tamaño no tiene por qué ser el mismo del array original) QUE CONTENGA SOLO LOS NÚMEROS QUE NO COINCIDEN CON EL ENTERO N**. REALIZA UN PROGRAMA JAVA QUE HAGA USO DE LA FUNCIÓN (main()). VISUALIZAR EL CONTENIDO DEL ARRAY ANTES DE LA LLAMADA, y DESPUES DE LA LLAMADA. (arraySinNumero.java)

Ver pag 128 a 133

CLASE Arrays – pag 133

SEGUIR EPIGRAFES DEL LIBRO

EJERCICIO ARRAY SUELDOS.

EJERCICIO ARRAYS ALUMNOS, ASIGNATURAS, NOTAS

HACER EJERCICIOS DEL LIBRO.

## 2. INTRODUCCIÓN A LOS ARRAYS BIDIMENSIONALES.

Para declarar una tabla o array bidimensional de F filas y C columnas escribimos lo siguiente:

```
TipoDato NombreArray[][];
```

```
NombreArray = new TipoDato[F][C];
```

Por ejemplo, para declarar una tabla o array de 3 filas y 2 columnas de enteros con estos valores escribimos lo siguiente:

	Columna 1	Columna 2
Fila 1	4	5
Fila 2	34	32
Fila 3	23	51

```
int F = 3, C = 2;
int Numeros[][] = new int[F][C];

Numeros[0][0] = 4;
Numeros[0][1] = 5;
Numeros[1][0] = 34;
Numeros[1][1] = 32;
Numeros[2][0] = 23;
Numeros[2][1] = 51;

System.out.println(Arrays.toString(Numeros[0]));
System.out.println(Arrays.toString(Numeros[1]));
System.out.println(Arrays.toString(Numeros[2]));
```

Visualiza:

```
[4, 5]
[34, 32]
[23, 51]
```

//RELLENAR MATRIZ en la declaración

```
int t[][] = {
    {1, 2, 3, 4},
    {5, 6},
    {7, 8, 9, 10}
};

System.out.println(Arrays.deepToString(t));
[[1, 2, 3, 4], [5, 6], [7, 8, 9, 10]]
```

**Recorrido de la matriz, con dos bucles**, uno para filas y otro para columnas, por ejemplo visualizo los elementos de la matriz, uno por uno:

```
for (int fila = 0; fila < F; fila++) {
    for (int columna = 0; columna < C; columna++) {
        System.out.printf("%2d ", Numeros[fila][columna]);
    }
    System.out.println();
}
```

**Muestra array bidimensional:**

```
System.out.println(Arrays.deepToString(Numeros));
```



```
[[4, 5], [34, 32], [23, 51]]
```

**Muestra el número de filas del array bidimensional y el número de columnas de la fila 0:**

```
System.out.println("Número de filas del array bidimensional: " + Numeros.length );
System.out.println("Número de columnas de la fila 0: " + Numeros[0].length );
```

**Método que devuelve una matriz de **f** filas y **c** columnas con números aleatorios:**

```
private static int[][] RellenarAleatorio(int f, int c) {
    int M[][] = new int[f][c];

    for (int fila = 0; fila < f; fila++) {
        for (int columna = 0; columna < c; columna++) {
            M[fila][columna] = (int) (Math.random() * (100) + 1);
        }
    }
    return M;
}
//
```

### Ejercicios arrays Bidimensionales:

1. Crear una matriz de 4x4, llenarla de 0 y la diagonal principal con 1.
2. Escribe la función “EscribeArray3x3” a la que le pasas un array bidimensional de tamaño 3x3 y te lo muestra por pantalla (que quede bonito).
3. Escribe la función “Rellena3x3” a la que le pasas una array bidimensional de 3x3 y te lo rellena con los números del 1 al 9.
4. Escribe la función “SumaArray3x3” a la que le pasas un array de 3x3 y te suma todos los números. Te devuelve un entero con el resultado.
5. Escribe la función “RellenaEnordenBi” a la que le pasas un array bidimensional de cualquier tamaño y te lo rellena con los números a partir del 1 en orden.
6. Escribe la función “SumaArrayBi” a la que le pasas un array bidimensional y calcula la suma por cada fila y columna y la muestra. Ejemplo para un array de 2 x 3.

1	2	3	--Suma:	6
4	5	6	--Suma:	15
-----				
5	7	9		

7. Escribe la función “RellenaArrayAjedrez” que te rellena un array de 8x8 con la forma de un tablero de ajedrez, usando el valor 1 para las casillas blancas y 0 para las negras.
8. Escribe una función que reciba dos matrices de dimensión NxN (pasa a la función la dimensión), y devuelva otra matriz de dimensión NxN con el resultado de la **suma**.  
Si las matrices **A**=(**a<sub>ij</sub>**) y **B**=(**b<sub>ij</sub>**) tienen la misma dimensión, la matriz suma es: **A+B**=(**a<sub>ij</sub>+b<sub>ij</sub>**). La **matriz suma** se obtiene sumando los elementos de las dos matrices que ocupan la misma misma posición. La matriz resta restando los elementos:

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 3 & 0 & 0 \\ 5 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$A+B = \begin{pmatrix} 2+1 & 0+0 & 1+1 \\ 3+1 & 0+2 & 0+1 \\ 5+1 & 1+1 & 1+0 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 2 \\ 4 & 2 & 1 \\ 6 & 2 & 1 \end{pmatrix}$$

$$A-B = \begin{pmatrix} 2-1 & 0-0 & 1-1 \\ 3-1 & 0-2 & 0-1 \\ 5-1 & 1-1 & 1-0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -2 & -1 \\ 4 & 0 & 1 \end{pmatrix}$$

Si la matriz no es cuadrada, no se puede sumar.

Ejemplo de salida:

```

MATRIZ 1:
=====
50      4      14
76      1      27
87      75     98

MATRIZ 2:
=====
60      59     62
68      86     77
73      96     75

MATRIZ SUMA:
=====
110     63     76
144     87    104
160    171    173
    
```

También se puede mostrar así:

MATRIZ 1			MATRIZ 2			SUMA		
=====								
50	4	14	.	60	59	62	.	110 63 76
76	1	27	.	68	86	77	.	144 87 104
87	75	98	.	73	96	75	.	160 171 173

9. Escribe una función que reciba dos matrices de dimensión NxN (pasa a la función la dimensión), y devuelva otra matriz de dimensión NxN con el resultado de la **multiplicación**. Ejemplo de producto de matrices de dimensión 2 y de dimensión 3:

<b>DIMENSIÓN 2:</b> $\begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} -1 & -1 \\ 5 & 6 \end{pmatrix} =$ $= \begin{pmatrix} 2 \cdot (-1) + 0 \cdot 5 & 2 \cdot (-1) + 0 \cdot 6 \\ 1 \cdot (-1) + 3 \cdot 5 & 1 \cdot (-1) + 3 \cdot 6 \end{pmatrix} =$ $= \begin{pmatrix} -2 & -2 \\ 14 & 17 \end{pmatrix}$	<b>DIMENSIÓN 3:</b> $\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix} \begin{pmatrix} 1 & -1 & 2 \\ 2 & -1 & 2 \\ 3 & -3 & 0 \end{pmatrix} =$ $= \begin{pmatrix} 1+8+21 & -1-4-21 & 2+8+0 \\ 2+10+24 & -2-5-24 & 4+10+0 \\ 3+12+27 & -3-6-27 & 6+12+0 \end{pmatrix} =$ $= \begin{pmatrix} 30 & -26 & 10 \\ 36 & -31 & 14 \\ 42 & -36 & 18 \end{pmatrix}$
--	--

<https://www.matesfacil.com/matrices/resueltos-matrices-producto.html>

## EJERCICIOS LIBRO 5.15