




2. Array multidimensionales

 Date	Empty
 Status	Not started
 Type	Empty
Units	4. Estructuras de almacenamiento

Arrays multidimensionales

- Acceso a los elementos
- Modificar valor de una posición
- Recorrer elementos
- Realizar búsquedas

Arrays multidimensionales

Los arrays multidimensionales se pueden definir como una colección de colecciones de objetos o array de arrays. Esto quiere decir que dentro de cada una de las posiciones del array se encontrará a su vez otro array.

Un array multidimensional está definido por dos parámetros, siendo el primero el número de posiciones y el segundo la posición concreta

primero el número de posiciones, y el segundo la posición concreta dentro del array de la posición indicada en el primer parámetro. En su creación estos dos elementos indican en número de filas y el número de columnas de cada fila.

```
//Los dos ejemplos representan el mismo tipo de array String
[][] multidimensional = {{ "primera", "posición"}, {"segunda", "p
osición"}, {"tercera", "posición"} }; String[][] multidimensiona
l = new String[2][3]; int[][] t1 = {{1,2,3}, {4,5,6}}; //2 fi
las por 3 columnas int[][] t2 = new int[2][3] //igual que en
el caso anterior, en el caso de tener objetos de diferentes t
ipos se utilizará array multidimensional de Object Object[][]
multidimensional = new Object[2][2];
```

Acceso a los elementos

Para poder acceder a un elemento concreto del array se realiza mediante posiciones.

```
int[][] t1 = {{1,2,3}, {4,5,6}}; //2 filas por 3 columnas t1
[1][1] // obtendrá el valor 4
```

Como se puede comprobar, el primer número indica la fila a la que se accede y el segundo indica la columna a la que se accede

al igual que pasa en la mayoría de casos en informática, el 0 es la primera posición

Modificar valor de una posición

Del mismo modo que en los arrays unidimensionales, también se puede asignar un valor a una posición

```
int[][] t1 = {{1,2,3},{4,5,6}}; System.out.println(t1[0][0]);
t1[0][0]=20; System.out.println(t1[0][0]);
```

Recorrer elementos

En el caso de querer recorrer el array se utiliza un for anidado para poder recorrer tanto las filas como las columnas

```
int[][] t1 = {{1,2,3},{4,5,6}}; System.out.println(t1[0][0]);
t1[0][0]=20; System.out.println(t1[0][0]); for (int i=0;i<t1.
length;i++){ System.out.printf("Array número %d",i); System.o
ut.println(); for (int j = 0; j<t1[i].length;j++){ System.ou
t.printf("Elemento del array %d",t1[i][j]); System.out.printl
n(); } }
```

O bien utilizar un foreach donde el primer bucle obtiene la fila y el segundo la columna

```
String[][] multidimensional = {"primera","posición"}, {"segun
da","posición"}, {"tercera","posición"}; int i = 0; for (Stri
ng[] temp : multidimensional){ System.out.println("El array q
ue se está evaluando es: "+temp.toString()); for (String temp
Dos: temp){ System.out.println("El valor es: "+tempDos); } }
```

Realizar búsquedas

Para poder buscar elementos dentro de un array multidimensional tan solo hay que recorrer los array al igual que se ha explicado, evaluando cada uno de los elementos de forma individual

```
int[][] t1 = {{1,2,3},{4,5,6}}; System.out.println(t1[0][0]);
t1[0][0]=20; System.out.println(t1[0][0]); for (int i=0;i<t1.
length;i++){ System.out.printf("Array número %d",i); System.o
ut.println(); for (int j = 0; j<t1[i].length;j++){ System.ou
t.printf("Elemento del array %d",t1[i][j]); System.out.printl
n(); if (t1[i][j]){ break; } } }
```