

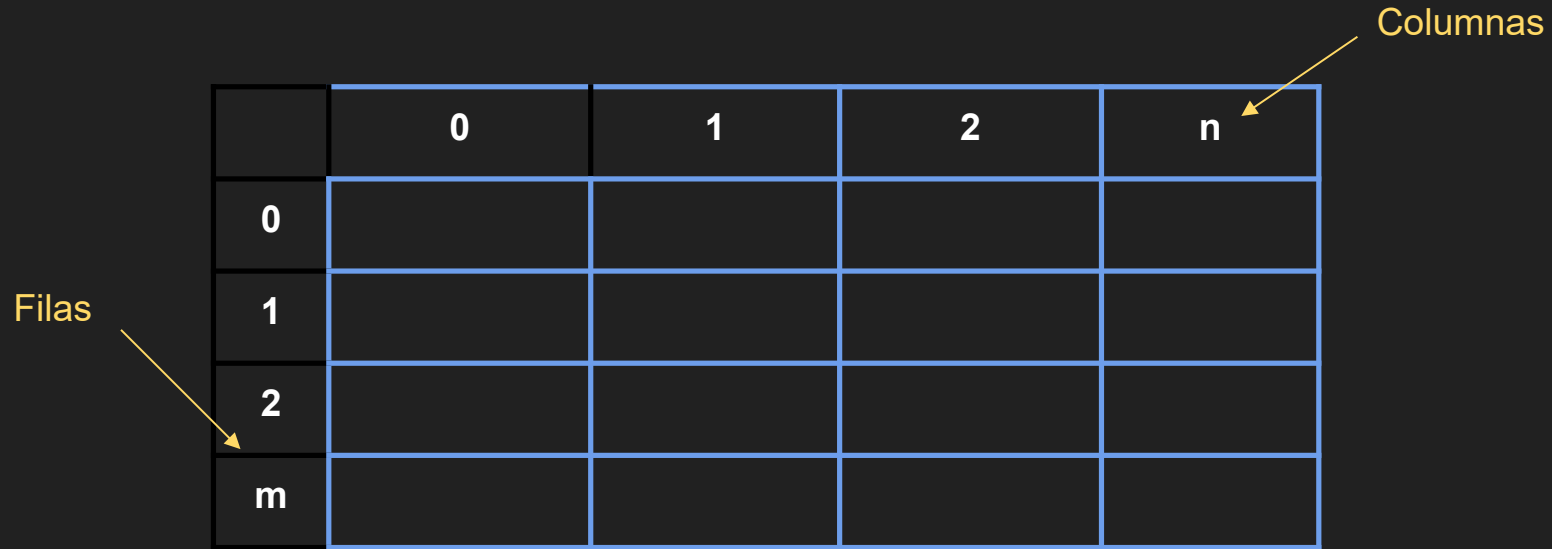
Unidad 2 - Temas 2

Arreglos Multidimensionales

Objetivos

Definir arreglos de dos o más dimensiones

Estructura de un array de dos dimensiones



The diagram illustrates the structure of a 2D array. It consists of a grid of cells. The first column is labeled 'Filas' (Rows) with an arrow pointing to it. The first row is labeled 'Columnas' (Columns) with an arrow pointing to it. The grid has 5 columns and 5 rows. The columns are indexed 0, 1, 2, and n. The rows are indexed 0, 1, 2, and m. The cells are empty, representing the data stored in the array.

	0	1	2	n
0				
1				
2				
m				

Declaración

<tipo de datoElemento> <nombre array> [][];

<tipo de datoElemento> [][]<nombre array>;

Ejemplos:

```
char pantalla[][];  
int puestos[][];  
double [][]matriz;
```

Inicialización

```
pantalla = new char[80][24]; // matriz con 80 filas y 24 columnas
puestos = new int[10][5];    // matriz de 10 filas por 5 columnas
final int N = 4;
matriz = new double[N][N];   // matriz cuadrada de N*N elementos
```

Operador new en la declaración

<tipo de datoElemento> <nombre array>[][]=

new <tipo de datoElemento> [<NúmDeFilas>][<NúmDeColumnas>];

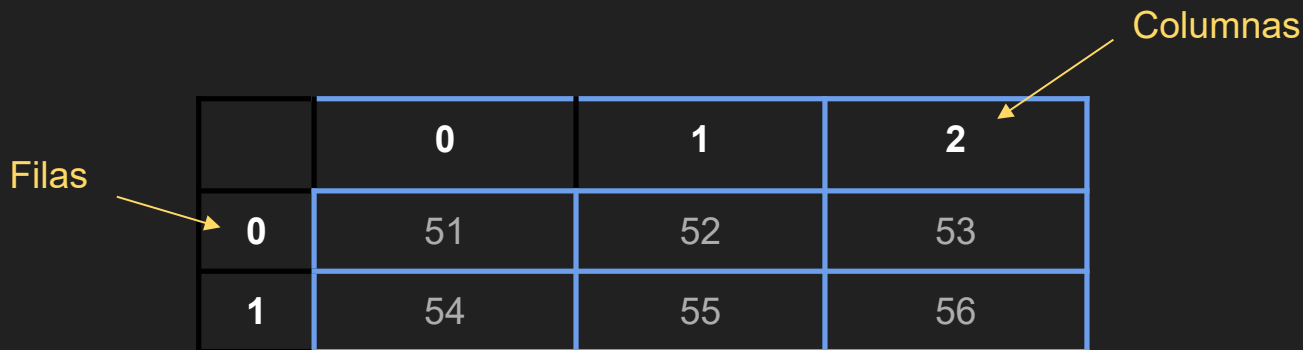


Cada dimensión
entre corchetes

Inicialización de arrays multidimensionales

```
int tabla1[][] = {{51, 52, 53},{54, 55, 56}};
```

```
int tabla1[][] = {  
    {51, 52, 53},  
    {54, 55, 56}  
};
```



The diagram illustrates a 2D array structure. It features a table with 3 columns and 2 rows. The first column is labeled 'Filas' (Rows) with an arrow pointing to the row index '0'. The second column is labeled 'Columnas' (Columns) with an arrow pointing to the column index '2'. The table contains the following values:

	0	1	2
0	51	52	53
1	54	55	56

Acceso a los elementos de arrays bidimensionales

Inserción de elementos

```
<nombre array>[indice fila][indice columna] = valor  
elemento;
```

Extracción de elementos

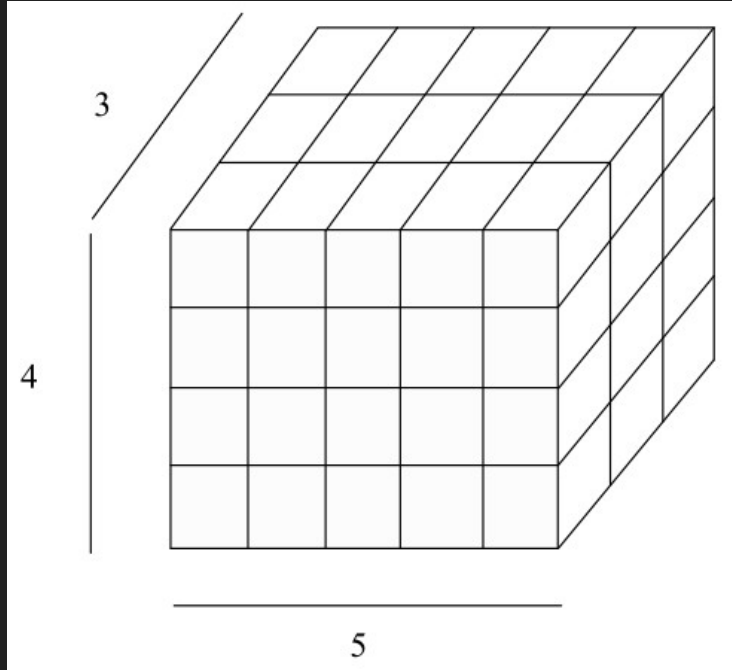
```
<variable> = <nombre array> [indice fila][indice columna];
```

Recorrido de una matriz bidimensional

```
for (fila = 0; fila < Matriz.length; ++fila){  
    for (col = 0; col < Matriz[fila].length; ++col){  
        Matriz[fila][col]  
    }  
}
```


Arreglos de más de dos dimensiones

```
int equipos[][][] = new int[4][5][3];
```



Actividad 1

Definir un array de 3 filas, cada una con un número aleatorio de columnas (entre 1 y 5). Cada celda debe ser llenada con un número aleatorio entre 1 y 100.

Se debe mostrar el resultado de la siguiente forma:

```
a[0][<cols>] = {12, 34, 56}
```

```
a[1][<cols>] = {12}
```

```
a[2][<cols>] = {12, 56}
```

Actividad 2

1. Realizar un programa que genere una tabla de dimensiones aleatorias $m \times n$ entre 1 y 5. Llenar la tabla de números aleatorios entre 1 y 100. Imprimir la tabla creada utilizando el atributo `Tabla.length` para determinar el tamaño de su dimensión. Debe mostrar de la siguiente forma

[0] = {12, 34, 67. 89}

[1] = {89, 34, 1, 23}

[2] = {...}

Bibliografía

AGUILAR, Luis Joyanes; ZAHONERO MARTÍNEZ, Ignacio. (2008). Estructuras de Datos en Java. España: McGraw-Hill. Pág. 69-75