

MATRICES EN JAVA

Charly Cimino

Matrices en Java

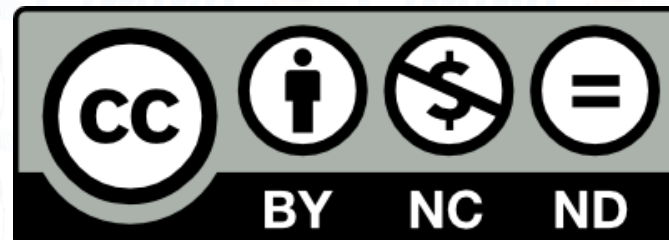
Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



Definición

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

En **matemática**, una matriz es un conjunto de números distribuidos en filas y columnas.

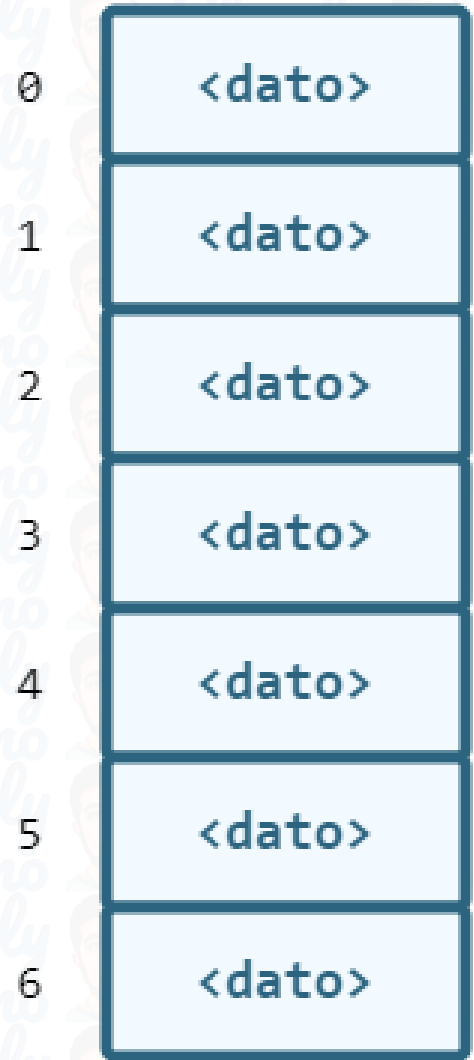
En **informática**, podemos representar una matriz de números, o cualquier otro tipo de dato, como un array de dos dimensiones.

- Array de arrays
- Array bidimensional
- Matriz (a secas)



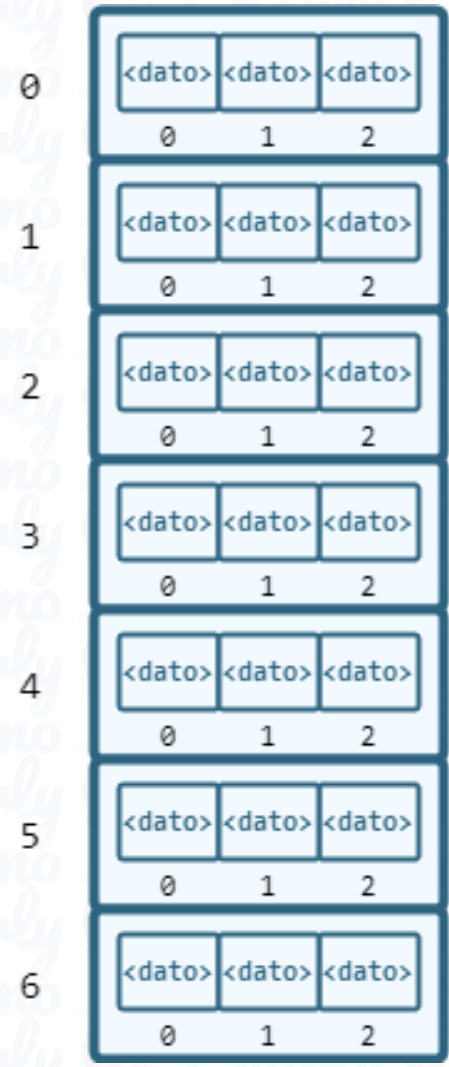
Sinónimos

Array unidimensional



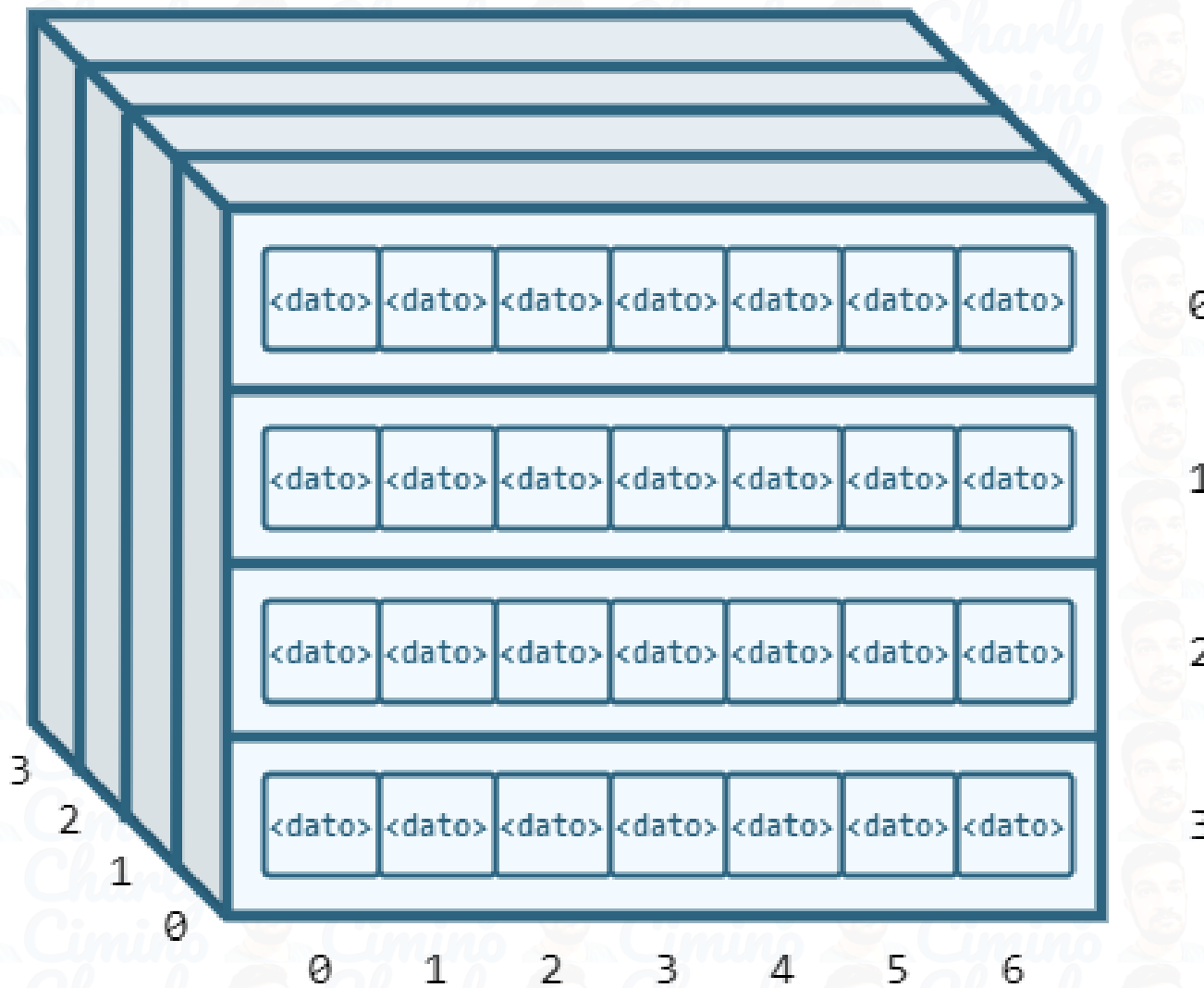
En cada celda del array hay un dato.

Array bidimensional



En cada celda del array hay un array unidimensional.

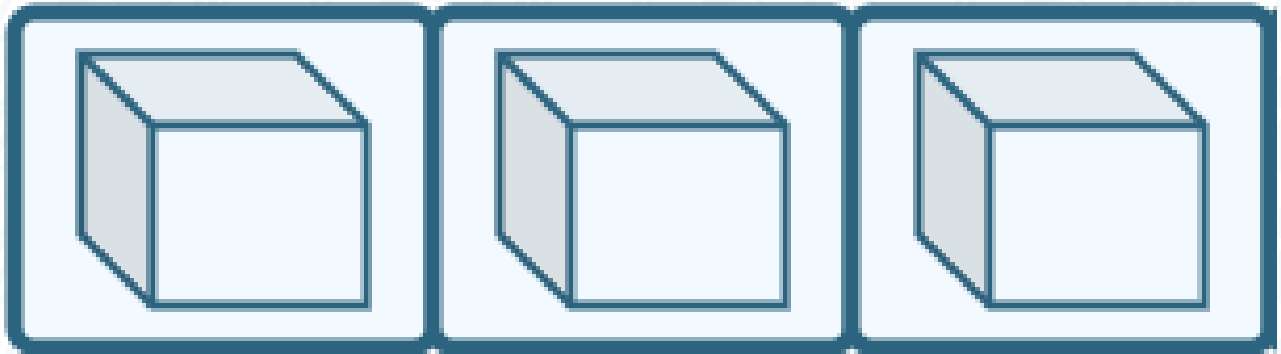
Array tridimensional



Arrays multidimensionales



Array cuatridimensional



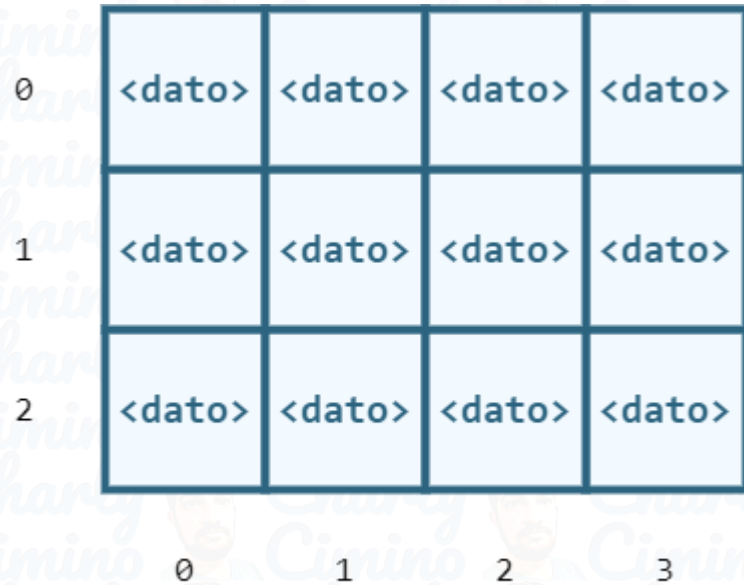
0

1

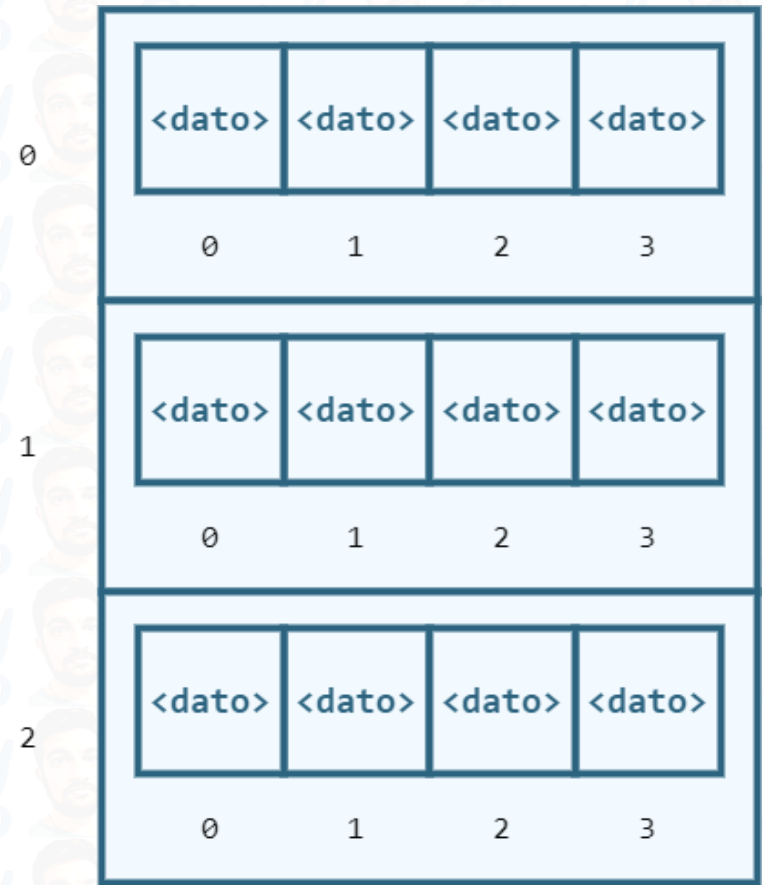
2

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

Volviendo a matrices...



Expectativa



Realidad

Convención

Matriz

=

Array de arrays

Tipo de dato 'matriz'

Para poder trabajar con variables que apunten a matrices, se debe usar el tipo de adecuado. Se pueden tener matrices de 'lo que sea' añadiendo un doble par de corchetes a la declaración.

```
int valor; // Un entero
int[] vecValores; // Un array de enteros
int[][] matValores; // Una matriz de enteros
```

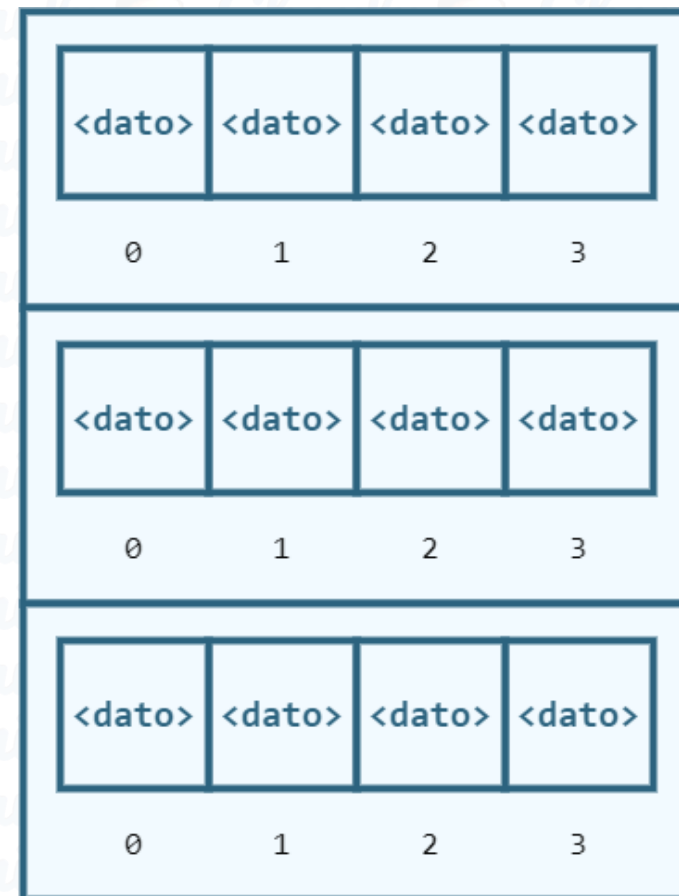
Crear una matriz

Como cualquier array, debe usarse el operador **new** para crear una nueva instancia.

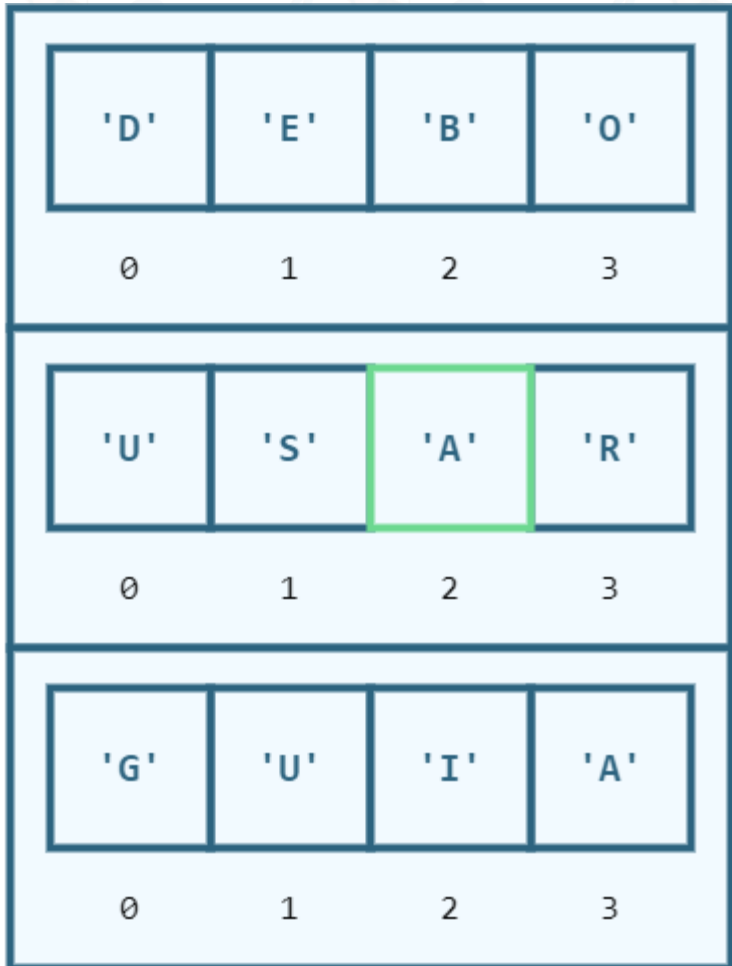
Cada par de corchetes representa una **dimensión**, cada una con su longitud.

```
Auto[][] matAutos = new Auto[3][4];
```

F C

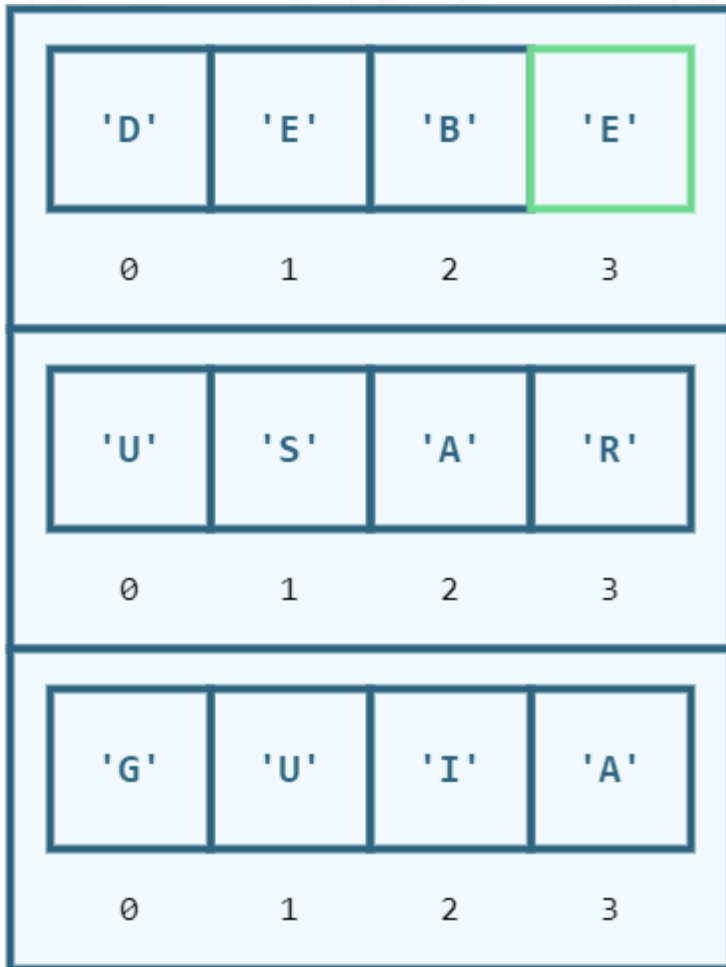


Obtener un valor



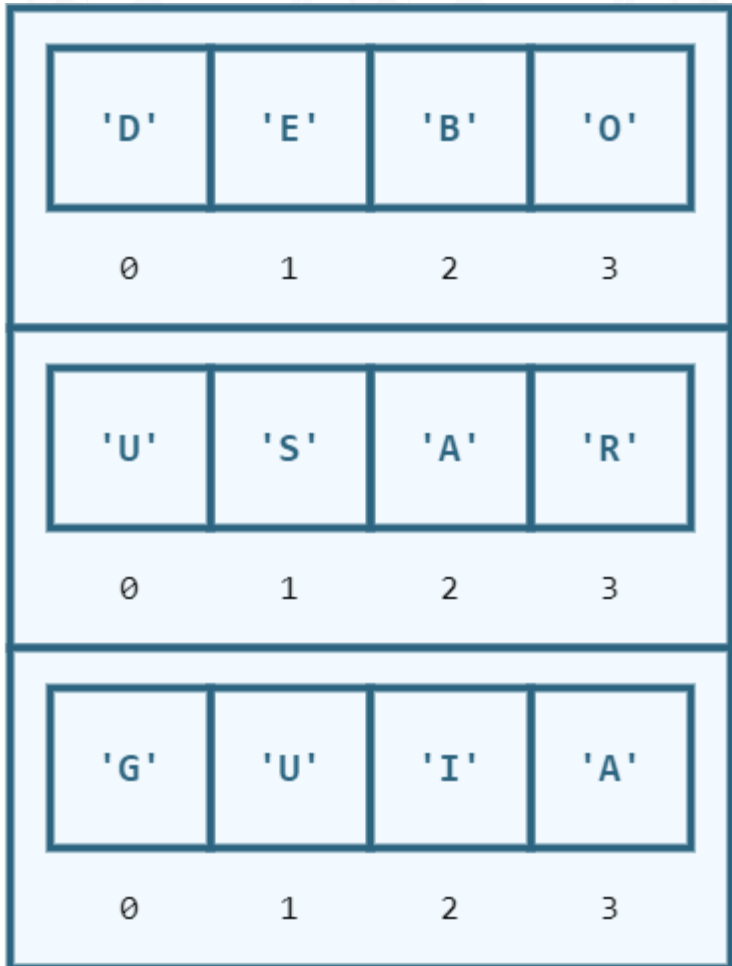
```
char[][] matLetras = {{'D', 'E', 'B', 'O'},  
                      {'U', 'S', 'A', 'R'},  
                      {'G', 'U', 'I', 'A'}};  
  
System.out.println( matLetras[1][2] ); // Muestra una 'A'
```

Establecer un valor



```
char[][] matLetras = {{'D', 'E', 'B', 'O'},  
                      {'U', 'S', 'A', 'R'},  
                      {'G', 'U', 'I', 'A'}};  
matLetras[0][3] = 'E'; // Establece una 'E'
```

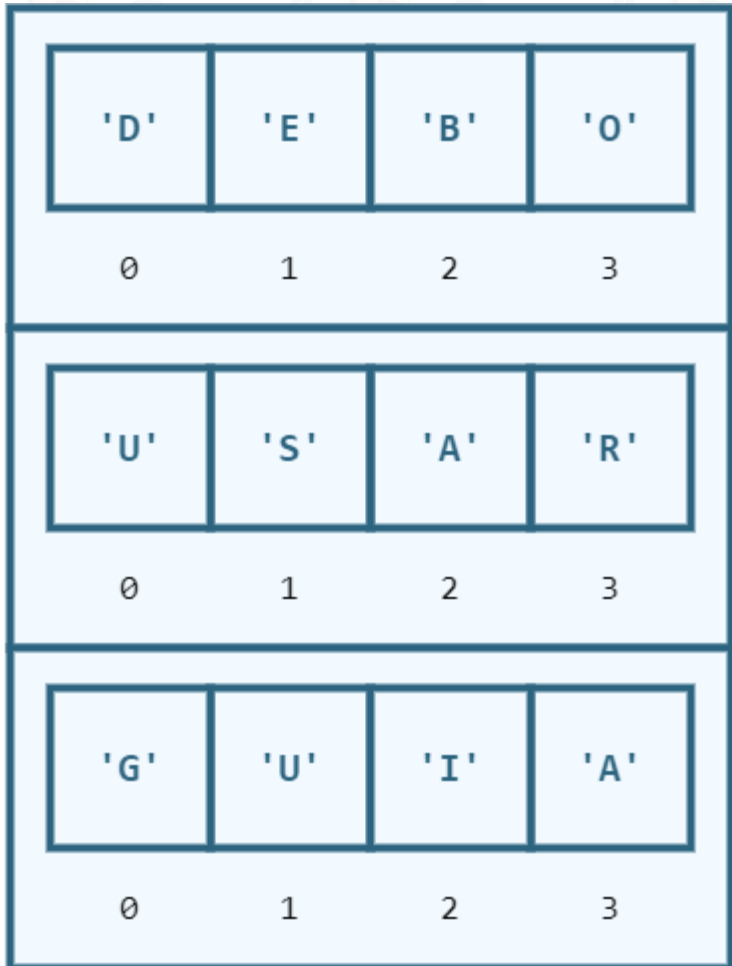

Obtener longitud



```
char[][] matLetras = {{'D', 'E', 'B', 'O'},  
                      {'U', 'S', 'A', 'R'},  
                      {'G', 'U', 'I', 'A'}};  
System.out.println( matLetras.length ); // Muestra 3
```

Recordar: es un array de arrays.

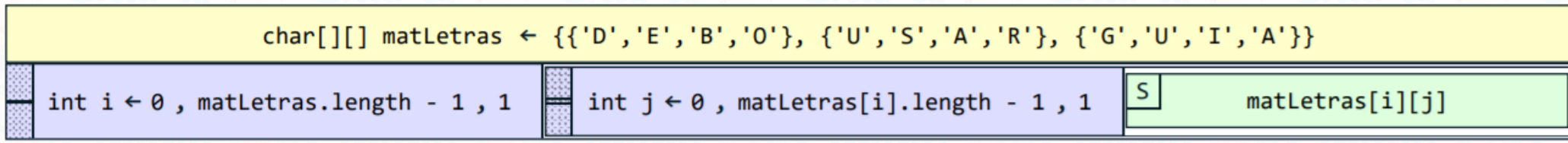
Obtener longitud



Asumimos que para ser una matriz, la longitud de cada uno de los arrays internos es la misma.

```
char[][] matLetras = {{'D', 'E', 'B', 'O'},  
{'U', 'S', 'A', 'R'}, {'G', 'U', 'I', 'A'}};  
int cant = matLetras.length * matLetras[0].length;  
System.out.println( cant ); // Muestra 12
```

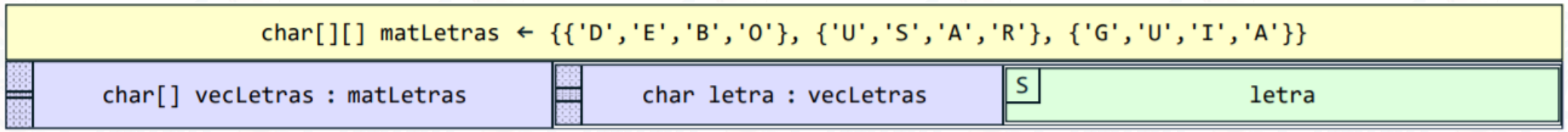
Recorrer todos los elementos



Usando ciclo for

```
char[][] matLetras = {{'D','E','B','O'}, {'U','S','A','R'}, {'G','U','I','A'}};  
for (int i = 0; i < matLetras.length; i++) {  
    for (int j = 0; j < matLetras[i].length; j++) {  
        System.out.println(matLetras[i][j]);  
    }  
}
```

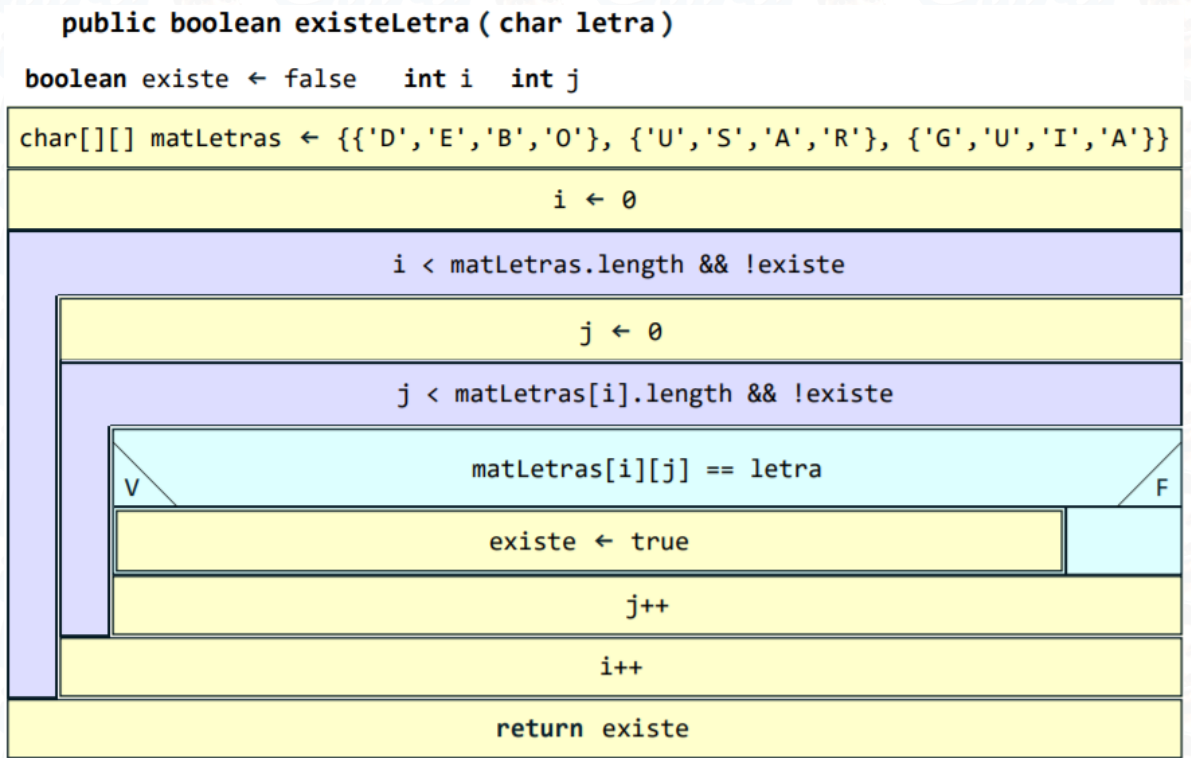
Recorrer todos los elementos



Usando ciclo foreach

```
char[][] matLetras = {{'D','E','B','O'}, {'U','S','A','R'}, {'G','U','I','A'}};
for (char[] vecLetras: matLetras) {
    for (char letra: vecLetras) {
        System.out.println(letra);
    }
}
```

Búsqueda en matriz



```

public boolean existeLetra (char letra) {
    char[][] matLetras = {{'D','E','B','O'},
                          {'U','S','A','R'},
                          {'G','U','I','A'}};

    boolean existe = false; int i, j;
    i = 0;
    while(i < matLetras.length && !existe) {
        j = 0;
        while(j < matLetras[i].length && !existe) {
            if(matLetras[i][j] == letra)
                existe = true;
            j++;
        }
        i++;
    }
    return existe;
}

```

FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).