

JAVA BASICO



Instructor: LENIN LEON

Email: leninleon0720@gmail.com

Celular: 6234-5608



ARREGLOS MULTIDIMENSIONALES (MATRICES)

También conocidas como matrices, son un tipo especial de vectores, funcionan de forma parecida que un vector, de igual forma almacenan elementos de un mismo tipo, el acceso a sus elementos se lo hace utilizando índices igual que en los vectores.

La diferencia fundamental es que en las matrices se maneja el concepto de fila y columna y por esa razón se las trabaja de forma diferente.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

COMO DECLARAR UNA MATRIZ

La declaración de una matriz se la puede hacer con la siguiente sentencia:

```
tipo_dato    nombre_matriz    [][]=    new    tipo_dato  
[dimension_filas][dimension_columnas];
```

Declarar una Matriz en Java

```
int miMatriz[][]= new int[5][2];//ejemplo declaración e  
inicialización
```



ARREGLOS MULTIDIMENSIONALES (MATRICES)

En donde:

- `int` es el tipo de dato para los elementos de la matriz, esta matriz sólo puede almacenar valores de tipo entero.
- `[] []` los 2 corchetes le indican al compilador que no estoy declarando una variable, ni un vector, si no una matriz.
- `miMatriz` es el nombre que se le da a la matriz, con este nombre podré acceder a su contenido y utilizar cuantas veces sea necesario.
- el operador `=` le dice al compilador que ese vector debe apuntar a una zona de memoria que se va reservar para la matriz.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

- la palabra reservada `new` crea un espacio en memoria para la matriz y la palabra `int` le dice que será de tipo entero.
- `[5] [2]` el número entre corchetes le indica que la dimensión que tendrá la matriz, el primer corchete indica el número de filas y el segundo corchete indica el número de columnas; es decir tendrá 5 filas y 2 columnas.

...



ARREGLOS MULTIDIMENSIONALES (MATRICES)

COMO LLENAR UNA MATRIZ

Existen 2 formas de llenar una matriz

La primera es dándole valores al momento de crearla (quemando los elementos), la sintaxis sería como sigue:
`tipo_dato nombre_matriz [][] = {{elemento1, elemento2, elemento3}, {elemento4, elemento5, elemento6}};`

Ejemplo como llenar una matriz quemando los elementos

```
int miMatriz [][] = {{2,5,3},{3,6,9},{0,8,9}};
```



ARREGLOS MULTIDIMENSIONALES (MATRICES)

En el caso anterior hemos declarado y llenado una matriz con elementos fijos, la dimensión de esta matriz es de 3×3 .

La llaves externas definen la matriz, para separar cada fila lo hacemos con llaves{} internas seguido de una coma (,) .

Para separa las columnas se utiliza una coma (,) después de cada elemento.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

La segunda forma es declararla, darle dimensiones y luego llenarla utilizando las posiciones de acuerdo a la siguiente sintaxis:

`nombre_matriz[posfila][poscolumna]=elemento;`



ARREGLOS MULTIDIMENSIONALES (MATRICES)

```
int miMatriz [][] =new int [3][3];  
    //llenar la primera fila  
    miMatriz[0][0]=5;  
    miMatriz[0][1]=8;  
    miMatriz[0][2]=1;  
    //llenar la segunda fila  
    miMatriz[1][0]=9;  
    miMatriz[1][1]=7;  
    miMatriz[1][2]=2;  
    // llenar la tercera fila  
    miMatriz[2][0]=10;  
    miMatriz[2][1]=15;  
    miMatriz[2][2]=25;
```



ARREGLOS MULTIDIMENSIONALES (MATRICES)

Como podemos observar las matrices al igual que los vectores empiezan desde la posición cero (0), es por eso que siempre para llenar o acceder a un elemento, debemos tener en cuenta de no acceder a posiciones que no existen.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

La forma anterior que muestra como llenar una matriz es un poco tedioso ya que si fueran muchos elementos tendríamos que repetir líneas de código, una buena forma de llenar una matriz es hacerlo un poco más dinámico utilizando 2 ciclos for, uno para manejar las filas y otro para las columnas.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

```
Scanner entrada = new Scanner(System.in);  
for (int i = 0; i < miMatriz.length; i++) {  
    for (int j = 0; j < miMatriz.length; j++) {  
        System.out.print("Ingresa un  
                           elemento: ");  
        miMatriz[i][j]=entrada.nextInt();  
    }  
}
```



ARREGLOS MULTIDIMENSIONALES (MATRICES)

En este caso se ha llenado pidiendo elementos desde la consola, en donde los indices corresponden a las variables i e j que llenan respectivamente cada posición de la matriz..



ARREGLOS MULTIDIMENSIONALES (MATRICES)

COMO ACCEDER A LOS ELEMENTOS DE UNA MATRIZ

De la misma forma que se puede llenar una matriz utilizando las posiciones vista en el ejemplo anterior, también se puede acceder a sus elementos utilizando las posiciones.

```
tipo_dato    variable=    nombre_matriz    [pos_fila]  
[pos_columna] ;  
int var=miMatriz[0][2];  
int var=miMatriz[0][2];
```



ARREGLOS MULTIDIMENSIONALES (MATRICES)

Si continuamos con el ejemplo anterior el valor que se almacena en la variable `var` es el número 1, puesto que está en las posiciones fila 0 columna 2 de la matriz.

Otra forma de acceder a los elementos de una matriz es recorrerla utilizando ciclos `for`, en este caso vamos a imprimir los elementos de la matriz del ejemplo anterior.



ARREGLOS MULTIDIMENSIONALES (MATRICES)

```
for (int i = 0; i < miMatriz.length; i++) {  
    for (int j = 0; j < miMatriz.length; j++) {  
        System.out.print(String.format(" %d ",miMatriz[i][j]));  
    }  
    System.out.println();  
}
```



Laboratorio 1

Ejemplo: Ingresar elementos en una matriz de 3×4 y obtener e imprimir la suma y el promedio por cada fila de la matriz.

