



# UNIVERSIDAD SANTO TOMÁS

## PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

---

### SECCIONAL TUNJA

---

VIGILADA MINEDUCACIÓN - SNIES 1732



Acreditación Institucional  
**Internacional**

OTORGADA POR EL IAC CINDE ACUERDO 55 DEL 8 DE MAYO-VIGENCIA 5 AÑOS



ACREDITACIÓN  
INSTITUCIONAL  
DE ALTA CALIDAD  
MULTICAMPUS

Res. MEN No. 03495 del 29 de enero de 2010  
Vigencia por seis años



QS STARS  
RATED FOR EXCELLENCE



ISO 9001  
Acreditación



CERTIFIED  
IONet  
MANAGEMENT SYSTEM





UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

**Faculty:** Systems engineer  
**Course:** Introduction of Programming  
**Topic:** Arreglos multidimensionales en JAVA

---

**Socializer:** Luis Fernando Castellanos Guarín  
**Email:** [Luis.castellanosg@usantoto.edu.co](mailto:Luis.castellanosg@usantoto.edu.co)  
**Phone:** 321-4582098



# Topics

## Arreglos

- **About / Acerca de**
- **Como funciona /How it works**
- **Sintaxis / Syntax**
- **Ejemplos / examples**
- **Como recorrer un array multidimensional**
- **Ejercicios / Exercises**

¡Siempre  
hacia lo alto!





# Arreglos Multidimensionales

Un array **multidimensional** son más de un array **unidimensional** unidos, para que te hagas una idea, es como si fuera una tabla. Se crea igual que un array unidimensional, solo hay que añadir un corchete mas con un tamaño en la definición del array

¡Siempre  
hacia lo alto!





## Syntax & Example

La sintaxis para declarar e inicializar un array multidimensional será:

```
Tipo_de_variable [ ] [ ] ... [ ] Nombre_del_array = new Tipo_de_variable[dimensión1][dimensión2]...[dimensiónN];
```

También podemos alternativamente usar esta declaración:

```
Tipo_de_variable [ ] [ ] ... [ ] Nombre_del_array;
```

```
Nombre_del_array = new Tipo_de_variable[dimensión1][dimensión2]...[dimensiónN];
```

### Ejemplo:

- `byte[][] edad = new byte[4][3];`
- `short [][] edad = new short[4][3];`
- `int[][] edad = new int[4][3];`
- `long[][] edad = new long[4][3];`
- `float[][] estatura = new float[3][2];`
- `double[][] estatura = new double[3][2];`
- `boolean[][] estado = new boolean[5][4];`
- `char[][] sexo = new char[2][1];`
- `String[][] nombre = new String[2][1];`



# Como recorrer un array multidimensional?

La sintaxis para declarar e inicializar un array multidimensional será:

```
Tipo_de_variable [ ] [ ] ... [ ] Nombre_del_array = new Tipo_de_variable[dimensión1][dimensión2]...[dimensiónN];
```

También podemos alternativamente usar esta declaración:

```
Tipo_de_variable [ ] [ ] ... [ ] Nombre_del_array;
```

```
Nombre_del_array = new Tipo_de_variable[dimensión1][dimensión2]...[dimensiónN];
```

**Ejemplo:**

- `byte[][] edad = new byte[4][3];`
- `short [][] edad = new short[4][3];`
- `int[][] edad = new int[4][3];`
- `long[][] edad = new long[4][3];`
- `float[][] estatura = new float[3][2];`
- `double[][] estatura = new double[3][2];`
- `boolean[][] estado = new boolean[5][4];`
- `char[][] sexo = new char[2][1];`
- `String[][] nombre = new String[2][1];`

¡Siempre  
hacia lo alto!



## How it works

3	8	2	1	6
---	---	---	---	---

//Array de 5 elementos

```
int[ ] variable1 = {3, 8, 2, 1, 6};
```

	0	1	2	3	4
matriz { 0					
1					
2					

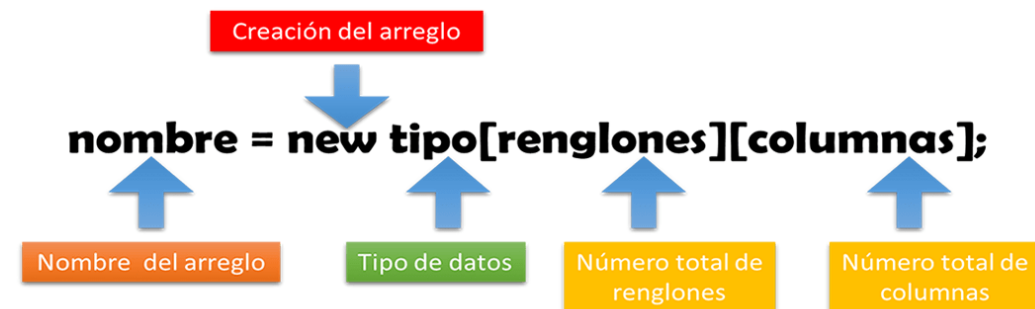
Esta matriz representa un arreglo bidimensional de **3 filas** por **5 columnas** cuyos valores son de tipo **int**.

La manera de declarar este arreglo sería:

```
int array [3][5];
```

//Definimos un array de 3 filas x 5 columnas

```
array[][]={{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
```





# How it works

```
public static void main(String[] args) throws IOException {  
74  
75     int[][] arreglo = new int[5][4];  
76  
77     arreglo[5][4] = 12;  
78  
79  
80  
81  
82  
}
```

1. El arreglo se declara de 5 filas, 4 columnas

2. Se intenta acceder a la dirección fila = 5, columna = 4

3. Se genera la excepción al intentar acceder a una dirección de fila fuera de los límites

Output - SumaMatrices2b17 (run) x

run:  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5  
at sumamatrices2b17.SumaMatrices2b17.main(SumaMatrices2b17.java:77)  
\\NetBeans\\Cache\\8.1\\executor-snippets\\run.xml:53: Java returned: 1  
BUILD FAILED (total time: 3 seconds)

**arreglo[fila][columna] = valor;**

Nombre del arreglo

Dirección del  
renglón

Dirección de  
la columna

Valor a almacenar  
en esa casilla





# To practice!

# LET'S GO!

¡Siempre  
hacia lo alto!



## P2Txx: Estudents\_grades

**create a JAVA software that:**

Store in a matrix the number of students (10) with which a subject has, with their respective four notes (P1: 20%, P2:25%, P3:25%, P4:30%.... Notes are random values between 0.0 and 5.0):

We will have 20 rows representing each student and 4 columns in which the respective notes will appear (0 x P1, 1 x P2, 2 x P3 and 3 x P4). It is requested to make the statement of the matrix and assign sample values to each item.

Construir un programa en JAVA que,

Almacene en una matriz el número de alumnos (10) con el que cuenta una asignatura, con sus respectivas cuatro notas (P1: 20%, P2:25%, P3:25%, P4:30%....Las notas son valores aleatorios entre 0.0 y 5.0):

Tendremos 20 filas que representarán a cada estudiante y 4 columnas en las que figurarán las respectivas notas (0 = P1, 1 = P2, 2 = P3 y 3 = P4). Se pide realizar la declaración de la matriz y asignarle unos valores de ejemplo a cada elemento.



## P2Txx: Estudents\_grades

		Notas del semestre (4)			
Estudiantes (10)		2	2.5	3	4.3
		4	3.7	3.2	3.8
		4.9	4.3	3.6	4.1
		2.3	3.7	4.4	4
		3.1	3.6	3.9	4
		5	4.8	4.6	4.7
		3.2	3	2.9	4
		2	2.3	2.1	3.6
		4	3.1	3.6	3
		3	3.9	4.1	4.7

```
Double [][] Notas = new double [10][4];
```

```
Notas [0][0] = 2;  
Notas [0][1] = 2.5;  
Notas [0][2] = 3 ;  
Notas [0][3] = 4.3;
```

```
Notas [1][0] = 4 ;  
Notas [1][1] = 3.7 ;  
Notas [1][2] = 3.2  
Notas [1][3] = 3.8;
```

```
Notas [2][0] = 4.9;  
Notas [2][1] = 4.3;  
Notas [2][2] = 3.6;  
Notas [2][3] = 4.1;
```

.....

¡Siempre  
hacia lo alto!





## P2Txx: Estudiantes\_grades

La mejor forma de recorrer el arreglo multidimensional es usando los ciclos FOR:

```
//Recorremos la primera columna de todas las filas
```

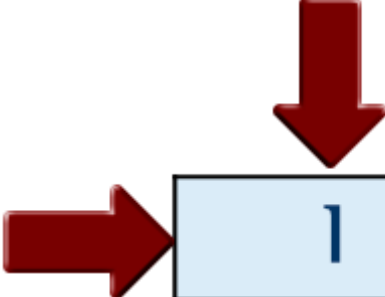
```
for(int i=0;i<array.length;i++){  
    System.out.println(array[i][0]);  
}
```

```
//Recorremos todas las columnas de la segunda fila
```

```
for(int i=0;i<array[0].length;i++){  
    System.out.println(array[1][i]);  
}
```

```
//Recorremos el array multidimensional
```

```
for (int i=0;i<array.length;i++){  
    for(int j=0;j<array[0].length;j++){  
        System.out.println(array[i][j]);  
    }  
}
```



1	2	3
4	5	6
7	8	9

< d\_codinGames />

¡Siempre  
hacia lo alto!



## P2Txx: swimming\_competition

### **create a JAVA software that:**

Allow to simulate the data of a swimming competition where the four times are stored for each competitor and also determine based on all the times of the competitors who is the winner. The user must specify how many times (competitors) he wants to enter and the times must be generated randomly between 6.0 and 10 sec per lap.

Additionally, you must view all the times of each competitor.

Construir un programa en JAVA que,

Permita simular los datos de una competencia de natación donde se almacenan los cuatro tiempos por cada competidor y además determinar con base en todos los tiempos de los competidores cual es el ganador. El usuario debe especificar cuantos tiempos (competidores) desea ingresar y los tiempos se deben generar de forma aleatoria entre 6.0 sec hasta 10 sec por vuelta.

Adicionalmente debe visualizar todos los tiempos de cada competidor.



## P2Txx: rating\_the\_food

### **create a JAVA software that:**

Let USTA learn how students rate food in the Giordano Bruno building cafeteria. For this, a scale of 1 to 10 was defined (1 denotes horrible and 10 denotes excellent). The software will perform a simulation where you randomly generate these grades for a month (20 days) for a number of students (the user will determine how many students will be surveyed).

With these ratings the software will determine how much was the percentage of those who rated the food as:

- bad ( $\leq 3$ )
- Regular (between 4 and 6)
- Excellent ( $\geq 7$ )

### Construir un programa en JAVA que,

Permita a la USTA conocer cómo califican los estudiantes la comida de la cafetería del edificio Giordano Bruno. Para ello definió una escala de 1 a 10 (1 denota horrible y 10 denota excelente). El software realizara una simulación donde genere de forma aleatoria dichas calificaciones durante un mes (20 días) para un número de estudiantes (el usuario determinara a cuántos estudiantes se encuestarán).

Con dichas calificaciones el software determinara cuanto fue el porcentaje de los que calificaron la comida como:

- mala ( $\leq 3$ )
- Regular (entre 4 y 6)
- Excelente ( $\geq 7$ )





## P2Txx: convert\_words

### Create a JAVA software that:

Allow to create a 4x5 matrix with the following information:

- maria, pedro, JOSE, camilo, rodrigo
- Ana, Lucia, Martha, Juliana, Patricia
- Felipe, ALEXANDER, soffy, Carmen, Augusto
- Alfredo, Luis, Ramiro, Karen, ANDREA

With the previous information, replace all the vowels with = a = @, e = 3, i = 1, o = Ø (alt + 157), u = ∩ (alt +127)

Convert all words to uppercase.

### Construir un programa en JAVA que,

Permita crear una matriz de 4x5 con la siguiente información:

- maria , pedro, JOSE, camilo, rodrigo
- Ana, lucia, martha, juliana, patricia
- Felipe, ALEXANDER, soffy, Carmen, Augusto
- Alfredo, Luis, Ramiro, Karen, ANDREA

Con la información anterior reemplazar todas las vocales por =a=@, e=3, i=1, o=Ø (alt+157) , u=∩ (alt +127)

Pasar todas las palabras a mayúsculas.



## P2Txx: numbers\_of\_rows

### Create a JAVA software that:

Fill a two-dimensional array of  $N \times M$  (values that the user must enter by keyboard), with random numbers between 1 and 1000.

Then the software will ask the user to enter a value between 1 to  $N$  (number of rows) and the software will display the sum, average, largest and smallest of the numbers that are in that row.

When the user enters a value equal to zero (0) it must be finished.

Construir un programa en JAVA que,

Llene un array bidimensional de  $N \times M$  (valores que debe ingresar por teclado el usuario), con números aleatorios entre 1 y 1000.

Luego el software pedirá al usuario que ingrese un valor entre 1 a  $N$  (número de filas) y el software mostrara la suma, el promedio, el mayor y el menor de los números que están en esa fila.

Cuando el usuario ingrese un valor igual a cero(0) se debe finalizar.



## P2Txx: vote\_for\_clubs\_and\_bars

### Create a JAVA software that:

Performs a simulation to vote on a bill in a department that proposes to end nightclubs and bars:

There are 123 municipalities and each municipality will vote for 0-NO, 1-YES (the votes will be random numbers between 0 and 5000).

Then the software must visualize which was the result of the vote: showing the winner and the loser with their percentage.

Additionally, you must view the rows that have the most voted for the winner and those that least vote for the loser.

### Construir un programa en JAVA que,

Realice una simulación para votar un proyecto de ley en un departamento que propone acabar con las discotecas y bares:

Son 123 municipios y cada municipio votara por el 0-NO , 1-SI (los votos serán números randomicos entre 0 y 5000).

Luego el software deberá visualizar cual fue resultado de la votación: mostrando el ganador y el perdedor con su porcentaje.

Adicional deberá visualizar las filas que tienen más votaron por el ganador y las que menos votaron por el perdedor.





## P2Txx: Estudents\_grades

### **create a JAVA software that:**

Perform the simulation of a voting process for the Boyacá governorate where there are seven (7) candidates from the main political parties in Colombia:

0-liberal, 1-Conservador, 2- Alianza verde, 3- La U, 4- cambio radical, 5-Centro democrático, 6-MIRA, 7-Polo.

Generation of the vote for the 123 municipalities: bearing in mind that the votes must be between 1 and 1000 but cannot be repeated.

In turn, get the total votes per party and determine who is the winner and who is second.

### Construir un programa en JAVA que,

Realice la simulación de un proceso de votación para la gobernación de Boyacá en donde estén siete (7) candidatos de los principales partidos políticos de Colombia:

0-liberal, 1-Conservador, 2- Alianza verde, 3- La U, 4- cambio radical, 5-Centro democrático, 6-MIRA, 7-Polo.

Genera la votación para los 123 municipios: teniendo presente que los votos deben estar entre 1 y 1000 pero no se pueden repetir.

A su vez obtener el total de los votos por partido y determinar quien es el ganador y quien es el segundo.







# UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

## ¡Siempre hacia lo alto!

[USTATUNJA.EDU.CO](http://USTATUNJA.EDU.CO)



@santotomastunja