

MATRICES

ARREGLOS BIDIMENSIONALES

CONTENIDO

- ❑ **Arreglos Bidimensionales (matrices)**
 - ❑ Qué son
 - ❑ Cómo se declaran
 - ❑ Cómo se insertan y leen datos
 - ❑ Ejemplos

MATRICES

También llamados arreglos bidimensionales

Es un conjunto de datos de un mismo tipo que están almacenados en arreglos de dos dimensiones.

Tienen una cantidad de filas y una cantidad de columnas

MATRICES

También llamados **arreglos bidimensionales**

Es un conjunto de datos de un mismo tipo que están almacenados en arreglos de dos dimensiones. Tienen una **cantidad de filas** y una **cantidad de columnas**

3.5	4.0	5.0	5.0
5.0	5.0	3.0	2.5
4.5	4.5	4.0	5.0

Arreglo bidimensional con 3 filas y 4 columnas

MATRICES

* Al igual que los arreglos unidimensionales, los índices empiezan a partir de cero y se indican entre corchetes: $[]$. El primer índice indica la **fila** y el segundo indica la **columna**.

					Columnas 0 a n-1			
Filas 0 a m-1					Elemento 0,0	Elemento 0,1	...	Elemento 0, n-1
					Elemento 1,0	Elemento 1,1	...	Elemento 1,n-1
					Elemento 2,0	Elemento 2,1	...	Elemento 2,n-1
				
					Elemento m-1,0	Elemento m-1,1	...	Elemento m-1,n-1

Matriz de mxn

ARREGLOS BIDIMENSIONALES

Índice para las
columnas

	0	1	2	3
0	3.5	4.0	5.0	5.0
1	5.0	5.0	3.0	2.5
2	4.5	4.5	4.0	5.0

Índice para las filas

PARA RECORDAR...

- * Una matriz almacena elementos del mismo tipo.
- * Una matriz es de tamaño fijo ($m \times n$).
- * Cada elemento se guarda en un espacio independiente.
- * Cada espacio se referencia con dos índices.
- * El primer índice referencia las filas.
- * El segundo índice referencia las columnas.
- * Los índices se empiezan a contar a partir de 0.
- * En una matriz de $m \times n$ sus índices irán de 0 a $m-1$ para las filas y de 0 a $n-1$ para las columnas.



MATRICES

Oscar	Sarah
Juan	Diana
Jhon	Andrea

Arreglo bidimensional 3 filas y 2 columnas

MATRICES

	0	1
0	Oscar	Sarah
1	Juan	Diana
2	Jhon	Andrea

Arreglo bidimensional 3 filas y 2 columnas

MATRICES

¿Es posible definir la siguiente matriz?

Sarah	24.8
Oscar	50.6
Kate	13.3

MATRICES

¿Es posible definir la siguiente matriz?

'D'	"Jhon"
"Juan"	'S'
"Oscar"	"Andrea"

MATRICES

- Cómo **definir** un arreglo bidimensional (Matriz)

MATRICES

- Cómo definir un arreglo bidimensional (Matriz)

tipoDeDato nombre[][]=new tipoDeDato[m][n];

donde **m** es la cantidad de filas y **n** es la cantidad de columnas

MATRICES

- `String nombres[][]=new String[3][2];`
- `double notas[][]=new double[50][4];`

MATRICES

- `String nombres[][]=new String[3][2];`

Arreglo bidimensional de Strings, llamado Nombres, con 3 filas y 2 columnas

- `double notas[][]=new double[50][4];`

Arreglo bidimensional de números reales, llamado Notas, con 50 filas y 4 columnas

nombres

	0	1
0	null	null
1	null	null
2	null	null

notas

	0	1	2	3
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
.				
.				
.				
.				
49	0.0	0.0	0.0	0.0

Declarando e Inicializando

Se puede declarar e inicializar una matriz al igual que los vectores:

Ejemplos:

```
String nombres[ ][ ] = {{ "Oscar", "Fonseca" },  
                          { "John" , "Santos"},  
                          { "Julio" , "Ruiz" }};
```

```
double notas[ ][ ] = {{ 5.0, 4.0, 5.0},  
                       { 4.3, 4.7, 3.8},  
                       { 2.7, 3.2, 4.0},  
                       { 4.3, 4.7, 5.0}};
```

Declarando e Inicializando

Se puede declarar e inicializar una matriz al igual que los vectores:

Ejemplos:

```
String nombres[ ][ ] = {{ "Oscar", "Fonseca" },  
                          { "John" , "Santos"},  
                          { "Julio" , "Ruiz" }};
```

Matriz de cadenas de texto de 3 filas y 2 columnas.

```
double notas[ ][ ] = {{ 5.0, 4.0, 5.0},  
                       { 4.3, 4.7, 3.8},  
                       { 2.7, 3.2, 4.0},  
                       { 4.3, 4.7, 5.0}};
```

Matriz de reales de 4 filas y 3 columnas.

MATRICES

Cómo **definir una matriz** de enteros, con 4 filas y 3 columnas, llamada **numeros**

MATRICES

Cómo definir una matriz de enteros, con 4 filas y 3 columnas, llamada `numeros`

```
int numeros[ ][ ] = new int[4][3];
```

MATRICES

3104567	24	109	-1
2134231	50	201	-30
1231141	13	130	-45
3123232	40	110	-63
5645343	23	150	-70
2233424	27	170	-5

¿Cómo definir este arreglo?

MATRICES

3104567	24	109	-1
2134231	50	201	-30
1231141	13	130	-45
3423232	40	110	-63
5645343	23	150	-70
2233424	27	170	-5

```
int datos[ ][ ]=new int[6][4]
```

MATRICES

3.104567	24.8
2.134231	50.6
1.231141	13.3
3.423232	40.5
5.645343	23.3
2.233424	27.5

¿Cómo definir esta matriz?

MATRICES

3.104567	24.8
2.134231	50.6
1.231141	13.3
3.423232	40.5
5.645343	23.3
2.233424	27.5

`double nombre[][]=new double[6][2];`

¿DÓNDE ESTÁ EL ERROR?

```
String nombres [] = new String[4][3];
```

```
int anchoLargo = new int [15][2];
```

```
double valorTiempo [][] = int [7,7];
```

```
String nombres [2][2] = {{"Oscar","100"},  
                           {"Sofia","A"}};
```

```
String nombres [][] = {{"Oscar","100"} {"Sofia","A"}};
```

```
String refs[][] = {{"a34" "b34" "d33"},  
                   {"cf2" "b12" "aa3"}};
```

MATRICES

¿Cómo **insertar** datos en los arreglos bidimensionales (matrices)?

MATRICES

¿Cómo **insertar** datos en los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna** donde va a almacenar el dato

nombreDelArreglo[**posicionFila**][**posicionColumna**]=valor;

MATRICES

¿Cómo insertar datos en los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna** donde va a almacenar el dato

```
nombres[0][0]="Oscar";
```

```
nombres[0][1]="Sarah";
```

nombres

Oscar	Sarah
null	null
null	null

MATRICES

¿Cómo **insertar datos** en los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna** donde va a almacenar el dato

```
nombres[0][0]="Oscar";  
nombres[0][1]="Sarah";  
nombres[?][?]="Jhon";  
nombres[?][?]="Andrea";
```

nombres

Oscar	Sarah
null	null
Jhon	Andrea

MATRICES

¿Cómo **insertar datos** en los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna** donde va a almacenar el dato

```
nombres[0][0]="Oscar";  
nombres[0][1]="Sarah";  
nombres[2][0]="Jhon";  
nombres[2][1]="Andrea";
```

nombres

Oscar	Sarah
null	null
Jhon	Andrea

MATRICES

¿Dónde puede haber errores?

```
int matriz[][] = new int [5][3];
```

```
double i;
```

```
int m=6,n=3;
```

...

```
matriz[0][3] = 21.2;
```

```
matriz[i][n] = 90;
```

```
matriz[m-1][n-1] = matriz [m][n] + 10;
```

```
matriz[5][3]=matriz[n][m];
```

...

MATRICES

¿Cómo recuperar los datos de los arreglos bidimensionales?

MATRICES

¿Cómo **recuperar los datos** de los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna**

nombreDelArreglo[**posicionFila**][**posicionColumna**]

MATRICES

¿Cómo **recuperar datos** de los arreglos bidimensionales?

Debe indicar la posición de la **fila** y de la **columna**

nombres[0][0]

nombres[0][1]

¿Cómo obtener "Andrea"?

nombres

Oscar	Sarah
null	null
Jhon	Andrea

MATRICES

- Presente el conjunto de instrucciones Java para **crear** una matriz de 50x4 números reales.
- Adicione las instrucciones necesarias para **solicitar** al usuario cada uno de los números
- Ahora, **muestre** en un mensaje, **todos** los números

MATRICES

- Presente el conjunto de instrucciones Java para **crear** una matriz de 50x4 números reales.

```
double numeros[][]= new double[50][4];
```

	0	1	2	3
0				
1				
2				

49				

MATRICES

- Presente el conjunto de instrucciones Java para **crear** una matriz de 50x4 números reales.
- Adicione las instrucciones necesarias para **solicitar** al usuario cada uno de los números

```
double numeros[][]= new double[50][4];
```

	0	1	2	3
0				
1				
2				

49				

	0	1	2	3
0				
1				
2				

49				

MATRICES

```
double numeros[][]= new double[50][4];  
numeros[0][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
```

	0	1	2	3
0				
1				
2				

49				

MATRICES

```
double numeros[][]= new double[50][4];  
  
numeros[0][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[0][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
  
numeros[1][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[1][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[1][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));  
numeros[1][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
```



```
numeros[0][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[0][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[0][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[0][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));

numeros[1][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[1][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[1][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[1][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
...

numeros[49][0]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[49][1]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[49][2]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
numeros[49][3]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero"));
```

```
for (int i=0; i<=49; i=i+1){
```

```
    for (int j=0; j<=3; j=j+1){
```

```
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero en  
la posición "+i+" "+j));
```

```
    }
```

```
}
```

```
for (int i=0; i<=49; i=i+1){ ←
```

La variable i maneja las filas.
Comienzan en 0, hasta 49

```
for (int j=0; j<=3; j=j+1){
```

```
    numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero en  
    la posición "+i+" "+j));
```

```
}
```

```
}
```

```
for (int i=0; i<=49; i=i+1){
```

```
for (int j=0; j<=3; j=j+1){ ←
```

La variable **j** maneja las columnas.
Comienzan en 0, hasta 3

```
    numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero en  
    la posición "+i+" "+j));
```

```
}
```

```
}
```



```
for (int i=0; i<=49; i=i+1){
```

```
for (int j=0; j<=3; j=j+1){
```

```
    numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite un numero de  
    la posición "+i+" "+j));
```

```
}
```

```
}
```

La variable **j** maneja las columnas.
Comienzan en 0, hasta 3

Se almacena cada número decimal solicitado en la posición **i,j** de la matriz

Matriz de 30x6

	0	1	...	5
0				
1				
2				

29				

```
For(int i=?; i<=?; i=i+1){  
    for(int j=?; j<=?; j=j+1){  
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite numero"));  
    }  
}
```

Matriz de 30x6

	0	1	...	5
0				
1				
2				

29				

```
For(int i=0; i<=29; i=i+1){
    for(int j=0; j<=5; j=j+1){
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite numero"));
    }
}
```

Matriz de 4x4

	0	1	2	3
0				
1				
2				
3				

```
For(int i=?; i<=?; i=i+1){  
    for(int j=?; j<=?; j=j+1){  
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite numero"));  
    }  
}
```

Matriz de 4x4

	0	1	2	3
0				
1				
2				
3				

```
For(int i=0; i<=3; i=i+1){  
    for(int j=0; j<=3; j=j+1){  
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite numero"));  
    }  
}
```

MATRICES

- Presente el conjunto de instrucciones Java para **crear** una matriz de 50x4 números reales.
- Adicione las instrucciones necesarias para **solicitar** al usuario cada uno de los números
- Ahora, **muestre** en un mensaje de texto, **todos** los números

```
double numeros = new double[50][4];  
for (int i=0; i<=49; i=i+1){  
    for (int j=0; j<=3; j=j+1){  
        numeros[i][j]=Double.parseDouble(JOptionPane.showInputDialog("Digite  
un numero"));  
    }  
}  
String mensaje="";  
for (int i=0; i<=49; i=i+1){  
    for (int j=0; j<=3; j=j+1){  
        mensaje=mensaje + numeros[i][j]+" ";  
    }  
    mensaje+="\n";  
}
```

Las matrices y el ciclo for

Una matriz se procesa generalmente usando dos ciclos for anidados:

```
int b[][] = new int[3][3];  
for ( int i = 0; i < 3; i++ ) {  
    for ( int j = 0; j < 3; j++ ) {  
        if (i == j)  
            b[i][j] = 1;  
        else  
            b[i][j] = 0;  
    }  
}
```


Las matrices y el ciclo for

Una matriz se procesa generalmente usando dos ciclos for anidados:

```
int b[][] = new int[3][3];  
for ( int i = 0; i < 3; i++ ) {  
    for ( int j = 0; j < 3; j++ ) {  
        if ( i == j )  
            b[i][j] = 1;  
        else  
            b[i][j] = 0;  
    }  
}
```

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

Las matrices y el ciclo for

`b.length` indica la cantidad de filas de la matriz.

`b[i].length` indica la cantidad de columnas de la matriz.

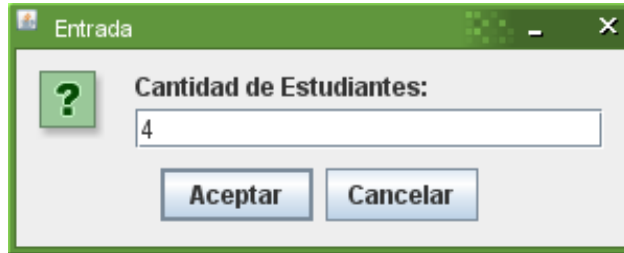
```
int b[][] = new int[3][3];
for ( int i = 0; i < 3; i++ ) {
    for ( int j = 0; j < 3; j++ ) {
        if ( i == j )
            b[i][j] = 1;
        else
            b[i][j] = 0;
    }
}
```

```
int b[][] = new int[3][3];
for ( int i = 0; i < b.length ; i++ ) {
    for ( int j = 0; j < b[i].length ; j++ ) {
        if ( i == j )
            b[i][j] = 1;
        else
            b[i][j] = 0;
    }
}
```

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

Ejemplo 1:

Escriba un programa en Java que solicite el código y el nombre de los estudiantes de cualquier curso y los muestre todos al final. Use una matriz para guardar los datos solicitados.



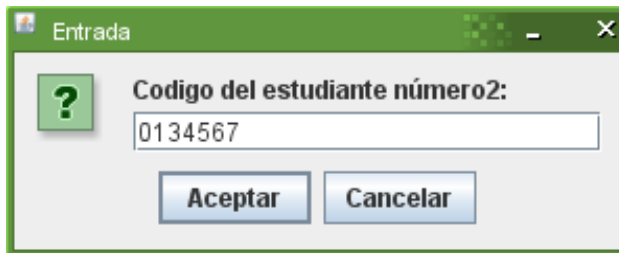
Entrada

?

Cantidad de Estudiantes:

4

Aceptar Cancelar



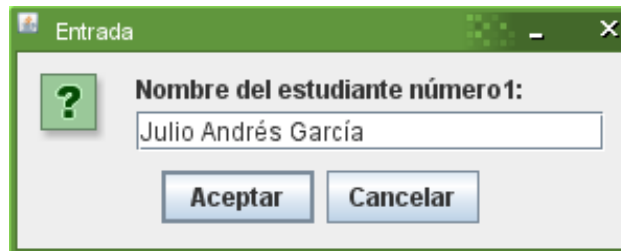
Entrada

?

Codigo del estudiante número2:

0134567

Aceptar Cancelar



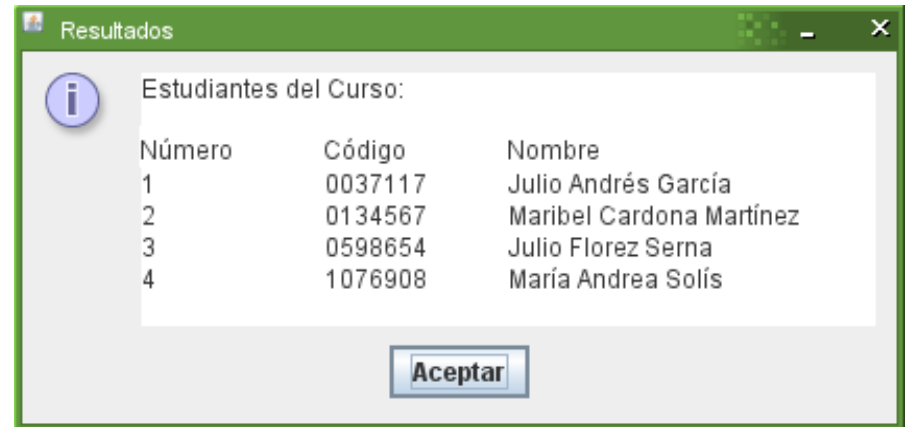
Entrada

?

Nombre del estudiante número1:

Julio Andrés García

Aceptar Cancelar



Resultados

i

Estudiantes del Curso:

Número	Código	Nombre
1	0037117	Julio Andrés García
2	0134567	Maribel Cardona Martínez
3	0598654	Julio Florez Serna
4	1076908	María Andrea Solís

Aceptar

Ejemplo 1: Análisis

- * Debemos capturar los nombres de m estudiantes. por lo tanto requerimos saber el valor de m para saber cuántas filas tendrá la matriz.
- * La matriz tendrá dos columnas: una para el código del estudiante (String) y otra para el nombre (String).
- * La matriz será del tipo String y de tamaño $m \times 2$.

"1035765"	"Julio Nava"
"0012345"	"Catalina Lee"
"1077890"	"Bruce Lee"

- * La salida del programa será un String que contendrá la lista numerada de los nombres de los estudiantes.

Ejemplo 1: Programa en Java

```
import javax.swing.*;

public class NombresCurso {

    static String estudiantesCurso[][];

    public static void main(String[] args) {
        int cantEstudiantes;
        String salida;

        cantEstudiantes = Integer.parseInt(
            JOptionPane.showInputDialog(
                "Cantidad de Estudiantes:"));

        estudiantesCurso = new String [cantEstudiantes][2];
    }
}
```

Ejemplo 1: Programa en Java

```
for (int m=0; m < estudiantesCurso.length; m++){
    estudiantesCurso[m][0]=JOptionPane.showInputDialog
        ("Codigo del estudiante número" + (m+1) + ":");

    estudiantesCurso[m][1]=JOptionPane.showInputDialog
        ("Nombre del estudiante número" + (m+1) + ":");
}

salida = "Estudiantes delCurso:\n\n" +
        "Número\tCódigo\tNombre\n";

for (int m=0; m < estudiantesCurso.length; m++){
    salida += (m+1) + "\t";
    for (int n=0; n < estudiantesCurso[m].length; n++){
        salida += estudiantesCurso[m][n] + "\t";
    }
    salida += "\n";
}
```

Ejemplo 1: Programa en Java

```
//mostrar los resultados en un Area de Texto
JTextArea areaSalida = new JTextArea();
JScrollPane scroll = new JScrollPane(areaSalida);
areaSalida.setText( salida );
JOptionPane.showMessageDialog( null, scroll,
    "Resultados", JOptionPane.INFORMATION_MESSAGE );

} //Fin método main
} //Fin clase
```

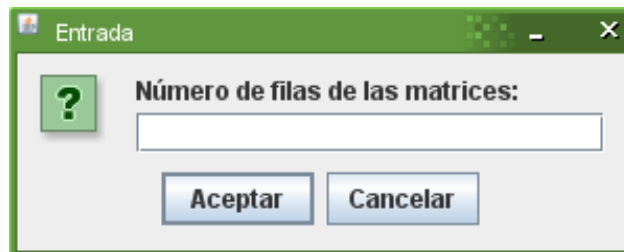
Ejemplo 2:

Escriba un programa que lea dos matrices de $m \times n$, calcule la suma de ellas y muestre el resultado.

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 3 & 0 & 0 \\ 5 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$A + B = \begin{pmatrix} 2+1 & 0+0 & 1+1 \\ 3+1 & 0+2 & 0+1 \\ 5+1 & 1+1 & 1+0 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 2 \\ 4 & 2 & 1 \\ 6 & 2 & 1 \end{pmatrix}$$

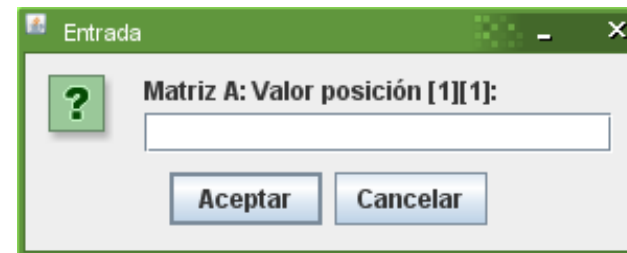
$$A - B = \begin{pmatrix} 2-1 & 0-0 & 1-1 \\ 3-1 & 0-2 & 0-1 \\ 5-1 & 1-1 & 1-0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -2 & -1 \\ 4 & 0 & 1 \end{pmatrix}$$



Entrada

? Número de filas de las matrices:

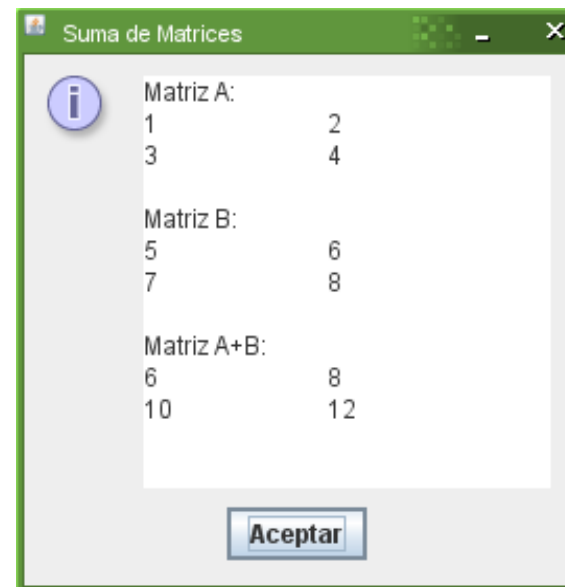
Aceptar Cancelar



Entrada

? Matriz A: Valor posición [1][1]:

Aceptar Cancelar



Suma de Matrices

i

Matriz A:

1	2
3	4

Matriz B:

5	6
7	8

Matriz A+B:

6	8
10	12

Aceptar

Ejemplo 2: Programa en Java

```
import javax.swing.*;

public class SumaMatrices {
    static String salida="";

    public static void main(String[] args) {

        int m, n,
        int a [][];
        int b [][];
        int c [][];

        m=Integer.parseInt(JOptionPane.showInputDialog(
            "Número de filas de las matrices:"));

        n=Integer.parseInt(JOptionPane.showInputDialog(
            "Número de Columnas de las matrices:"));

        a = new int [m] [n];
        b = new int [m] [n];
        c = new int [m] [n];
```

//Continúa...

Ejemplo 2: Programa en Java

```
//Continuación método main
```

```
leerMatriz(a, "Matriz A");
```

```
leerMatriz(b, "Matriz B");
```

```
calcularSuma(a,b,c);
```

```
generarSalida(a, "Matriz A");
```

```
generarSalida(b, "Matriz B");
```

```
generarSalida(c, "Matriz A+B");
```

```
JTextArea areaSalida = new JTextArea();
```

```
areaSalida.setText( salida );
```

```
JOptionPane.showMessageDialog( null, areaSalida,  
    "Suma de Matrices", JOptionPane.INFORMATION_MESSAGE );
```

```
}//fin método main
```

Ejemplo 2: Programa en Java

```
static void leerMatriz(int matriz[][], String s) {  
    for(int m=0; m < matriz.length; m++){  
        for(int n=0; n < matriz[m].length; n++){  
            matriz[m][n]=Integer.parseInt(  
                JOptionPane.showInputDialog(  
                    s + ": Valor posición [" +  
                    m + "][" + n + "]:");  
            }  
        }  
    }  
} //fin método leerMatriz
```

Ejemplo 2: Programa en Java

```
static void calcularSuma(int a[][], int b[][], int c[][]) {  
    for(int m=0; m < c.length; m++){  
        for(int n=0; n < c[m].length; n++){  
            c[m][n] = a[m][n] + b[m][n];  
        }  
    }  
} //fin método calcularSuma  
  
static void generarSalida (int a[][], String titulo) {  
    salida += titulo + ":\n";  
    for(int m=0; m < a.length; m++){  
        for(int n=0; n < a[m].length; n++){  
            salida += a[m][n] + "\t";  
        }  
        salida += "\n";  
    }  
    salida += "\n";  
} //fin método generarSalida  
} //fin clase
```

ARREGLOS BIDIMENSIONALES

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5

- Cómo mostrar en el área de texto solo los elementos de la primera fila

```
String salida="";
```

```
for (int i=0; i<=3; i++){
```

```
    salida=salida+ numeros[0][i] +"\n";
```

```
}
```

```
miArea.append(salida);
```

ARREGLOS BIDIMENSIONALES

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5


- Cómo mostrar en el área de texto solo los elementos **de la primera columna**

ARREGLOS BIDIMENSIONALES

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5

- Cómo mostrar en el área de texto los elementos de la **diagonal **


```
for (int i=0; i<=3; i++){  
    for (int j=0; j<=3; j++){  
        if (i==j){  
            miArea.append("\n" + numeros[i][j]);  
        }  
    }  
}
```



De todas las posiciones,
solo muestra los número,
cuando la fila es igual a la
columna (diagonal \)

ARREGLOS BIDIMENSIONALES

- Muestre la **suma** de todos los números en la matriz

```
int suma=0;
for (int i=0; i<=3; i++){
    for (int j=0; j<=3; j++){
        suma = suma + numeros[i][j];
    }
}
miArea.append("\n La suma es : " + suma )
```

ARREGLOS BIDIMENSIONALES

- Muestre la suma de los elementos de la diagonal \

```
int suma=0;
for (int i=0; i<=3; i++){
    for (int j=0; j<=3; j++){
        if (i==j){
            suma = suma + numeros[i][j];
        }
    }
}
miArea.append("\nLa suma es : " + suma )
```

ARREGLOS BIDIMENSIONALES

- Muestre la suma de los elementos de cada columna

ARREGLOS BIDIMENSIONALES

2	5	2
3	5	12
6	12	43
21	32	31

La suma de la columna 1 es: 32

La suma de la columna 2 es: 54

La suma de la columna 3 es: 88

```
int sumaCol;
for (int col=0; col<3; col++){
    sumaCol=0;
    for (int fil=0; fil<4; fil++){
        sumaCol = sumaCol + numeros[fil][col];
    }
    miArea.append("\nLa suma de la columna" + (col+1) + " es: " +
        sumaCol);
}
```


ARREGLOS BIDIMENSIONALES



Ejercicio: Se requiere una aplicación en java para almacenar los resultados de las ultimas elecciones de rector de la universidad del Valle. Los datos deben almacenarse en una matriz donde cada fila corresponde a una sede y cada columna corresponde a un candidato. El programa debe mostrar la tabla con los nombres de las sedes y los nombres de los candidatos y cada uno de los resultados. La aplicación también debe mostrar el candidato ganador.

Se debe mostrar en un JTextArea todos los valores del arreglo.

MATRICES



Arreglos a utilizar

Candidatos	" Ivan Ramos"	"Jorge Sanchez"	" José Rios"
------------	---------------	-----------------	--------------

Sedes	" Cali"	"Palmira"	"Buga"	"Tuluá"
-------	---------	-----------	--------	---------

500	400	300
250	150	250
200	250	100
120	200	210

Votos

MATRICES



Arreglos a utilizar

Candidatos	" Ivan Ramos"	"Jorge Sanchez"	" José Rios"
------------	---------------	-----------------	--------------

Sedes	" Cali"	"Palmira"	"Buga"	"Tuluá"
-------	---------	-----------	--------	---------

Ramos	Sanchez	Rios
↓	↓	↓
500	400	300
250	150	250
200	250	100
120	200	210

Votos

MATRICES



Arreglos a utilizar

Candidatos	" Ivan Ramos"	"Jorge Sanchez"	" José Rios"
------------	---------------	-----------------	--------------

Sedes	" Cali"	"Palmira"	"Buga"	"Tuluá"
-------	---------	-----------	--------	---------

		Ramos ↓	Sanchez ↓	Rios ↓
Cali	→	500	400	300
Palmira	→	250	150	250
Buga	→	200	250	100
Tuluá	→	120	200	210

Votos

ARREGLOS BIDIMENSIONALES



```
public class votaciones{  
    public static void main (String a[]) {  
        String candidatos[], sedes[];  
        int votos[][] , fil, col;  
        JTextArea area= new JTextArea (15, 30);;  
        JScrollPane scroll = new JScrollPane (area);  
        fil = Integer.parseInt(JOptionPane.showInputDialog  
                                ("Ingrese el número de sedes:"));  
  
        sedes = new String[fil];  
  
        for (int x = 0; x < fil; x++){  
            sedes[x] = JOptionPane.showInputDialog ("Sede No  
                                                    : "+ (x+1) );  
        }  
    }  
}
```

ARREGLOS BIDIMENSIONALES



```
col = Integer.parseInt(JOptionPane.showInputDialog(  
    "Ingrese el número de candidatos:"));  
  
candidatos = new String[col];  
  
for (int x = 0; x < col; x++){  
    candidatos[x] = JOptionPane.showInputDialog("Nombre del  
        Candidato No : "+(x+1));  
}  
  
votos = new int[fil][col];  
  
for (int x = 0; x < fil; x++){  
    for (int y = 0; y < col; y++){  
        votos[x][y] = Integer.parseInt(JOptionPane.  
            showInputDialog("Ingrese los votos de la sede  
                " +sedes[x]+ "para el candidato" +candidatos[y]));  
    }  
}
```

ARREGLOS BIDIMENSIONALES



```
int[] totalVotos = new int[candidatos.length];
for (int y = 0; y < col; y++){
    for (int x = 0; x < fil; x++){
        totalVotos[y] += votos[x][y];
    }
}

int max = 0;
int pos = 0;
for (int i = 0; i < totalVotos.length; i++){
    if ( max < totalVotos[i] ){
        max = totalVotos[i];
        pos = i;
    }
}

area.append("Candidato con mayoría de votos:
"+candidatos[pos]+" Total de Votos: "+totalVotos[pos]);
}
```

ARREGLOS BIDIMENSIONALES



```
for (int i = 0; i < fil; i++){
    area.append("\n"+sedes[i]);
    for (int x = 0; x < col; x++){
        area.append("\n"+candidatos[x]);
        area.append("\t"+votos[i][x]);
    }
}

JOptionPane.showMessageDialog(null, scroll);
} //fin main
} //fin clase
```