



**NOMBRE DE LA MATERIA: PROGRAMACIÓN**

**NRC: 42555**

**HORARIO: MARTES Y JUEVES 9 AM – 10:55AM**

**ESTUDIANTE: EFRAIN ROBLES PULIDO**

**CODIGO: 221350095**

**TEMA: ARREGLOS BIDIMENSIONALES**

**FECHA: 31 DE OCTUBRE DE 2021**

## Arreglos bidimensionales

	0	1	2	3	...	n
0						
1						
2						
3						
4						
...						
m						

Un arreglo de dos dimensiones equivale a una tabla con múltiples filas y múltiples columnas. Si las filas se etiquetan de 0 a m y las columnas de 0 a n, el número de elementos que tendrá el arreglo será el resultado del producto  $(m+1) \cdot (n+1)$ . El sistema para localizar un elemento será a través de las coordenadas representadas por sus número de fila y su número de columna (a, b).

La sintaxis para la declaración de un arreglo de dos dimensiones es:

```
<tipo de datoElemento> <nombre array> [<NúmeroDeFilas>] [<NúmeroDeColumnas>]
```

Algunos ejemplos:

```
char Pantalla[25][80];  
int puestos[6][8];  
int equipos[4][30];  
int matriz[4][2];
```

Entonces un arreglo de dos dimensiones en realidad es un arreglo de arreglos. Es decir, es un arreglo unidimensional, y cada elemento no es un valor entero, o de coma flotante o carácter, sino que cada elemento es otro arreglo.

Los elementos de los arreglos se almacenan en memoria de modo que el subíndice más próximo al nombre del arreglo es la fila y el otro subíndice, la columna. En esta tabla se representan todos los elementos y sus posiciones relativas en memoria del arreglo, `int tabla [4][2]`, suponiendo que cada entero ocupa 2 bytes.

Elemento	Posición relativa de memoria
tabla [0][0]	0
tabla [0][1]	2
tabla [1][0]	4
tabla [1][1]	6
tabla [2][0]	8
tabla [2][1]	10
tabla [3][0]	12
tabla [3][1]	14

### Inicialización de arreglos bidimensionales

Al igual que los de una dimensión, cuando se declaran. La inicialización consta de una lista de constantes separadas por comas y encerradas entre llaves, como en los ejemplos siguientes:

```
1. int tabla[2][3] = {51, 52, 53, 54, 55, 56};
```

o bien en los formatos más amigables:

```
int tabla[2][3] = { {51, 52, 53},  
                  {54, 55, 56} };  
int tabla[2][3] = {{51, 52, 53}, {54, 55, 56}};  
int tabla[2][3] = {  
    {51, 52, 53},  
    {54, 55, 56}  
};
```

```
2. int tabla2[3][4] = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

Tabla [2] [3]		0	1	2	Columna
Fila	0	51	52	53	
	1	54	55	56	

  

Tabla [3] [4]		0	1	2	3	Columna
Fila	0	1	2	3	4	
	1	5	6	7	8	
	2	9	10	11	12	

## Acceso a los elementos de los arreglos bidimensionales

Se puede acceder a los elementos de arreglos bidimensionales de igual forma que a los elementos de un arreglo unidimensional. La diferencia reside en que en los elementos bidimensionales deben especificarse los índices de la fila y la columna. El formato general para asignación directa de valores a los elementos es:

inserción de elementos

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

extracción de elementos

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

Algunos ejemplos de asignación de valores:

```
Tabla[2][3] = 4.5;  
Resistencias[2][4] = 50;  
AsientosLibres[5][12] = 5;
```

Y de extracción de valores:

```
Ventas = Tabla[1][1];  
Dia = Semana[3][6];
```

## Acceso a elementos mediante bucles

Se puede acceder a los elementos de arreglos bidimensionales mediante bucles anidados, el bucle externo para el acceso a las filas y el bucle interno para las columnas. Su sintaxis es:

```
int IndiceFila, IndiceCol;  
for (IndiceFila = 0; IndiceFila < NumFilas; ++IndiceFila)  
    for (IndiceCol = 0; IndiceCol < NumCol; ++IndiceCol)  
        Procesar elemento[IndiceFila][IndiceCol];
```

## Bibliografía:

Joyanes Aguilar, L., (2014). Programación en C, C++, Java y UML. McGraw Hill. 2a. Edición