



## Arreglos Bidimensionales: Matrices

Hasta el momento nuestros arreglos eran de una dimensión, y se utilizaban para representar listas:

[ ] - [ ] - [ ] - .... - [ ]

pero, es posible definir arreglos de dos dimensiones, o también conocidos como matrices:

[ ] - [ ] - [ ] - .... - [ ]

[ ] - [ ] - [ ] - .... - [ ]

.....

[ ] - [ ] - [ ] - .... - [ ]

Si bien parece una estructura compleja, un arreglo bidimensional es tratado en forma muy similar a un arreglo unidimensional.

## Declaración de Matrices

Var nombre: ARRAY [ rango1, rango2 ] OF Tipo de Dato

O

ARRAY[ Fila, Columna] OF Tipo de Dato

## Ejemplo

Var MATRIZ: ARRAY [1..3, 1..5] OF Integer;

|       |  | Columnas |     |       |     |       |
|-------|--|----------|-----|-------|-----|-------|
| Filas |  | 23       | 54  | 37566 | 235 | 556   |
|       |  | 1467     | 235 | 989   | 54  | -6765 |
|       |  | 9765     | 2   | -8965 | 22  | 7676  |

MATRIZ es un arreglo bidimensional de 3x5, esto significa que tiene 3 filas y 5 columnas.

Originalmente esta matriz debe estar vacía, si es una matriz numérica debemos asignarle el valor cero al inicializarla.

Esto nos asegura de que la memoria no contenga datos no deseados.

|       |  | Columnas |   |   |   |   |
|-------|--|----------|---|---|---|---|
| Filas |  | 0        | 0 | 0 | 0 | 0 |
|       |  | 0        | 0 | 0 | 0 | 0 |
|       |  | 0        | 0 | 0 | 0 | 0 |

Si queremos cargar un valor en una determinada posición solo hacemos: MATRIZ [columna, fila]:= valor

Por ejemplo: Matriz[2,2]:=3;

|       |  | Columns |          |   |   |   |
|-------|--|---------|----------|---|---|---|
| Filas |  | 0       | 0        | 0 | 0 | 0 |
|       |  | 0       | <b>3</b> | 0 | 0 | 0 |
|       |  | 0       | 0        | 0 | 0 | 0 |

El resto de las operaciones son similares a las de vectores.

Se puede utilizar variables para recorrer las filas y las columnas de manera de leer, mostrar o copiar una matriz. Veamos un ejemplo:

```
uses crt;
var Matriz: array[1..3, 1..4] of integer;
c, f: integer;
begin
  randomize;
  clrscr;
  // Rellena la matriz con números al azar
  for f:=1 to 3 do
    for c:=1 to 4 do Matriz[f,c] := random(10);
  { Imprime la matriz }
  for f:=1 to 3 do
    begin
      for c:=1 to 4 do
        write( '[', Matriz[f,c], ']' );
      writeln;
```

```
end;  
readln;  
end.
```

## Tipos de Acceso

**Acceso secuencial:** Se accede elemento a elemento, es necesario una exploración secuencial comenzando desde el primer elemento.

**Acceso directo o aleatorio:** Permite acceder a un elemento determinado haciendo una referencia directa por su posición en la estructura.

## Operaciones sobre Matrices

Las operaciones que se pueden realizar con matrices durante el proceso de resolución de un problema son:

- Inicializar
- Cargar / Actualizar (añadir, borrar, insertar)
- Mostrar
- Copiar

También se puede:

- Recorrer (acceso secuencial)
- Buscar