

## TEMA 5. Anexo Arrays Bidimensionales. Matrices

Los ARRAYS multidimensionales son arrays en los que necesitamos más de un índice para acceder a un elemento individual. Las dimensiones pueden ser las que se quiera, pero raramente excederán de tres y lo más común son los ARRAYS **bidimensionales**, también conocidos como **matrices**.

En este apartado vamos a estudiar las matrices. Nos servirán para definir tablas de datos de un mismo tipo.

La representación de una matriz se puede ver como una tabla de dos dimensiones, en la que la primera dimensión la representan las filas y la segunda las columnas. Por ejemplo, una matriz de 5 x 4 de números enteros se podría representar así:

```
int [,] matriz = new int[5, 4];
```

35	18	125	-36
-2	5	66	185
125	-3001	14	233
-3	333	17	18
0	45	39	100

Como hemos visto la sintaxis general para declarar y crear la matriz es:

```
tipo [,] nombre = new tipo[numFilas, numColumnas];
```

Al igual que ocurre con los vectores, en este caso el número de filas irá desde 0 hasta numFilas - 1 y el número de columnas desde 0 hasta numColumnas - 1

Para poder referenciar los elementos de un ARRAY multidimensional se coloca el nombre y luego tantos índices como dimensiones se tienen.

```
matriz[2, 3] = 233;
```

La primera dimensión, viéndolo como una tabla, se denomina filas, y la segunda columnas.

**Ejemplo. Lectura y escritura de una matriz de 3 filas y 4 columnas:**

```
const int kFILAS = 3;
const int kCOLS = 4;

// Declaramos y creamos la matriz con las dimensiones.
int[,] matriz = new int[kFILAS, kCOLS];

void leerMatriz(int[,] matriz)
{
    int i, j;

    // Recorremos los elementos de la matriz.
    for (i = 0; i < kFILAS; i++)
        for (j = 0; j < kCOLS; j++)
            matriz[i, j] = int.Parse(TextBox("Elemento[" + i + ", " + j
                + "]"));
}

string mostrarMatriz(int[,] matriz)
{
    int i, j;
    string texto;

    texto = "Los valores de la matriz son:\n";

    // Recorremos las filas
    for (i = 0; i < kFILAS; i++)
    {
        // Recorremos las columnas de cada fila
        for (j = 0; j < kCOLS; j++)
            texto = texto + matriz[i, j] + ", ";
        texto = texto + "\n"; // Retorno de carro al terminar cada fila
    }

    return texto;
}

private void bLeer_Click(object sender, EventArgs e)
{
    leerMatriz(matriz);
}

private void bMostrar_Click(object sender, EventArgs e)
{
    string texto;

    texto = mostrarMatriz(matriz);

    MessageBox.Show(texto);
}
```

La función `GetLength(Dimensión)` a semejanza de `Length` en los vectores devuelve el número de elementos de la dimensión de array pasada como parámetro.

Así `matriz.GetLength(0)` devolverá el número de filas, y `matriz.GetLength(1)` devolverá el número de columnas.

### Ejemplo:

Realizar un subprograma que recorra una matriz de enteros y obtenga el mayor número de la matriz. Realizar otra que obtenga el menor.

```
int numeroMayor(int[,] matriz)
{
    int mayor, i, j;

    // Al principio nos quedamos con el primer numero.
    mayor = matriz[0, 0];

    for (i = 0; i < matriz.GetLength(0); i++)
        for (j = 0; j < matriz.GetLength(1); j++)
            if (matriz[i, j] > mayor)
                mayor = matriz[i, j];

    return mayor;
}

int numeroMenor(int[,] matriz)
{
    int menor, i, j;

    // Al principio nos quedamos con el primer numero.
    menor = matriz[0, 0];

    for (i = 0; i < matriz.GetLength(0); i++)
        for (j = 0; j < matriz.GetLength(1); j++)
            if (matriz[i, j] < menor)
                menor = matriz[i, j];

    return menor;
}
```

<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/arrays/multidimensional-arrays>