

Las matrices o arrays (también llamados arreglos o vectores)

son estructuras que puede almacenar varios valores simultáneamente (siempre del mismo tipo). Cada uno de estos valores se identifica mediante un número al cual se llama índice. Así, para acceder al primer elemento del array habría que usar el índice cero, para el segundo el índice uno, para el tercero el índice dos, y así sucesivamente. Vamos a ver cómo se declara un array:

```
tipo[] variable; int[] edades;
```

Si sabemos desde el principio cuantos datos tenemos (por ejemplo 4), les reservaremos espacio con

```
int[] edades = new int[4];  
edades[0] = 18;  
edades[1] = 20;  
edades[2] = 17;  
edades[3] = 24;
```

Si sabemos desde el principio los valores de los datos, pondremos

```
int[] edades = new int[] {18, 20, 17, 24};
```

Por definición **un arreglo es de una medida fija y por lo tanto no puede ser extendido**, es decir no podemos redimensionarlo. Esto aplica para arreglos de cualquier tipo: int, string, char, etc.

Para arreglar esto lo haremos con listas, que ya lo veremos mas adelante
Referencia a los valores.

con array es muy común utilizar los bucles 'for'

```
for (i=0; i<=4; i++)  
Console.WriteLine(edades[i]);
```

Tamaños de los arreglos

```
int tamanoEdades = edades.Length;
```

Ejercicio: Hacer una aplicación de consola que muestre por pantalla el valor mínimo y máximo del array tamaño 5 donde previamente el usuario ha introducido los valores del array

Introduce valor 1: 45

Introduce valor 2: 35

Introduce valor 3: 5

Introduce valor 4: 36

Introduce valor 5: 89

El valor minimo es "5" y el máximo es '89'

//Ordena el array (Array.sort())

string[] cadenas = {"Laura", "Martín", "Pedro", "Eva"};

Array.Sort(cadenas);

//Recorre el array comenzando con el índice 0 y terminando con el índice Length-1

foreach (string texto in cadenas)

{

Console.WriteLine(texto);

}

Ejercicios propuestos:

- Un programa que pida al usuario 5 números reales y luego los muestre en el orden contrario al que se introdujeron.*
- Un programa que pida al usuario 10 números enteros y calcule (y muestre) cuál es el mayor de ellos.*

Arrays bidimensionales o mas

Podemos declarar arrays de dos o más dimensiones. Por ejemplo, si queremos guardar datos de dos grupos de alumnos, cada uno de los cuales tiene 20 alumnos:

```
int datosAlumnos[2,20]
```

y entonces sabemos que los datos de la forma

```
datosAlumnos[0,i]    //son los del primer grupo, y los
```

```
datosAlumnos[1,i]    //son los del segundo.
```

```
int[,] notas1 = new int[2,3]; // array entero de 2x3
notas1[0,0] = 1;
notas1[0,1] = 2;
notas1[0,2] = 3;
notas1[1,0] = 3;
notas1[1,1] = 4;
notas1[1,2] = 5;
```

```
int[,] notas2 = // el mismo que antes
{
{1, 2, 3},
{3, 4, 5}
};
```

Tamaños de los arreglos

`notas2.GetLength(0)` // Da el número de filas (la primera dimensión)

`notas2.GetLength(1)` // Da el número de columnas (la segunda dimensión)

arrays de arrays

```
int[][] notas = new int[3][];
```

```
notas [0] = new int[5];
```

```
notas [1] = new int[4];
```

```
notas [2] = new int[2];
```

Cada uno de los elementos es una matriz unidimensional de enteros. El primer elemento es una matriz de 5 enteros, el segundo es una matriz de 4 enteros y el tercero es una matriz de 2 enteros.

```
int[][] notas = new int[3][];
```

```
notas [0] = new int[] { 1, 3, 5, 7, 9 };
```

```
notas [1] = new int[] { 0, 2, 4, 6 };
```

```
notas [2] = new int[] { 11, 22 };
```

Ejercicio propuesto:

Crear un programa que tengamos dos arrays, uno que tengas los nombres de los códigos de una tienda y otro que tenga tanto el precio como el descuento
Los valores del primer array son

- *Camisa Caballero*
- *Camisa Señora*
- *Pantalón Caballero*
- *Pantalón Señora*

Los precios serán 25, 18, 33, 35 respectivamente y sus descuentos serán del 10% 5% 2% 0% respectivamente

Hacer un programa que te muestre los siguiente

- *Camisa caballero 22.50 €*
- *Camisa señora 17, 10 €*
- *Pantalón Caballero 32,34 €*
- *Pantalón Señora 35,00 €*