

# ESTRUCTURA DE DATOS

## UNIDAD II.- ARREGLOS

### Tema 2.1.- Arreglos Unidimensionales y Bidimensionales.

## ARREGLOS - ARRAYS

La creación y manipulación de arrays en C# es similar a la de cualquier otro lenguaje de programación, aunque difiere un poco en cuanto a sintaxis (desde la perspectiva de un programador en Java).

## ARREGLOS - CARACTERÍSTICAS

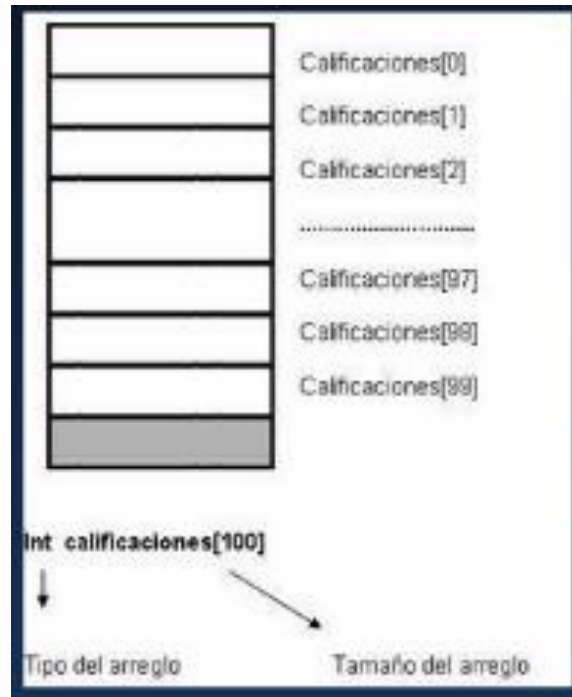
1. Un arreglo **está formado por un número fijo de elementos contiguos de un mismo tipo**. Al tipo se le llama **tipo base del arreglo**. Los datos individuales se llaman **elementos del arreglo**.
2. Para definir un tipo estructurado arreglo, **se debe especificar el tipo base y el numero de elementos**.

## ARREGLOS - CARACTERÍSTICAS

3. Un array se clasifica por:
- **Almacena los elementos** del array en posiciones de memoria continua.
  - Tener un **único nombre de variable** que representa a todos los elementos, y éstos a su vez se diferencian por un **índice o subíndice**.
  - **Acceso directo o aleatorio** a los elementos individuales del array.

# ARREGLOS - CLASIFICACIÓN

## 1. Unidimensionales (vectores o listas)



## 2. Multidimensionales (tablas o matrices)

	0	1	2	3
Fila 0	1	4	7	10
Fila 1	8	7	3	5
Fila 2	1	6	8	20

# ARREGLOS - CLASIFICACIÓN

## 1.- Unidimensionales

- *Es un tipo de dato estructurado que está formado de una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales.*
- El tipo de acceso a los arreglos unidimensionales es el *acceso directo*, es decir, podemos acceder a cualquier elemento del arreglo sin tener que consultar a elementos anteriores o posteriores, esto *mediante el uso de un índice para cada elemento* del arreglo que nos da su posición relativa.

# ARREGLOS - CLASIFICACIÓN

## 1.- Unidimensionales

- Para implementar arreglos unidimensionales se debe reservar espacio en memoria, y se debe proporcionar la dirección base del arreglo, **la cota superior y la inferior.**

# ARREGLOS - CLASIFICACIÓN

## 2.- Bidimensionales

- Este tipo de arreglo al igual que los anteriores es un tipo de dato estructurado, finito ordenado y homogéneo. El acceso a ellos también es en forma directa por medio de un par de índices.
- Los arreglos bidimensionales se usan para representar datos que pueden verse como una tabla con filas y columnas.



# ARREGLOS - CLASIFICACIÓN

## 2.- Bidimensionales

- Son **matrices o arreglos de dos dimensiones, cuyos elementos tienen dos índices**. En C# existen dos tipos: los **rectangulares** y los **dinámicos**. Por lo general, cuando accedemos a arreglos bidimensionales utilizamos los términos filas y columnas.
- En los arreglos **bidimensionales rectangulares**, cada fila tiene la misma cantidad de columnas. Por otro lado, las filas de los arreglos **bidimensionales dinámicos** pueden tener diferente cantidad de columnas.

## ARREGLOS - CLASIFICACIÓN

### 3.- Matriz Escalonada

- Es una matriz cuyos elementos son matrices. Los elementos de una matriz escalonada pueden ser de diferentes dimensiones y tamaños.
- Una matriz escalonada se denomina también "matriz de matrices".

## ARREGLOS – Utilización en C#

**Existen dos maneras básicas de iniciar un arreglo:**

1. Al asignarle un tamaño estamos iniciándolo con los valores por defecto (en el caso de enteros, es cero y para el caso de cadenas es null).

- **Matrices unidimensionales:**

Ejemplo:

```
int[] x;           // declarar el arreglo en este caso es de una dimensión.  
x = new int[ 10 ]; // Para asignarle un tamaño tenemos que crear el  
                  // arreglo asignándole un valor
```

- **Matrices multidimensionales:**

Ejemplo:

```
string[,] regiones; // declarar el arreglo en este caso es de dos dimensiones.  
regiones = new string[2,3]; // Para asignarle un tamaño tenemos  
                           // que crear el arreglo asignándole un valor
```

## ARREGLOS – Utilización en C#

- **Matriz escalonada(Matriz de matrices):**

### **Ejemplo 1:**

```
int[][] matrizEscalonada= new int[3][]; //Declarando la matriz
```

```
matrizEscalonada[0] = new int[5]; //Iniciando la matriz  
matrizEscalonada[1] = new int[4];  
matrizEscalonada[2] = new int[2];
```

### **Ejemplo 2. Método abreviado:**

```
int[,] jaggedArray4 = new int[3],  
{  
    new int[,] { {1,3}, {5,7} },  
    new int[,] { {0,2}, {4,6}, {8,10} },  
    new int[,] { {11,22}, {99,88}, {0,9} }  
};
```

## ARREGLOS – Utilización en C#

En el momento de la declaración podemos asignar directamente los valores, poniéndolos dentro de corchetes ({ ... }), separados por comas.

- **Matrices unidimensionales**

**Ejemplo:**

```
int[] x;
```

```
x = new int[ 10 ] { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
```

- **Matrices multidimensionales:**

**Ejemplo:**

```
string[,] regiones;
```

```
regiones = new string[2,3] { { "Argentina", "Brasil", "Peru",  
                               "USA", ";Mexico", "Costa Rica" } };
```

## ARREGLOS – Utilización en C#

- **Matriz escalonada (Matrices de matrices)**

Ejemplo 1:

//Declarando la matriz

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

//Inicializando la matriz

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

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

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

//Asignándole valores a cada matriz

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

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

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

## **ARREGLOS – Aplicación en C#**

Realiza los siguientes programas:

1. Realiza un programa que contenga un método que invierta un arreglo unidimensional con valores.
2. Realiza un programa que me diga si existe un elemento dentro de una arreglo de dos dimensiones.

## ARREGLOS – Aplicación en C#

**Para mas información y ejemplos visita:**

- <http://www.puntopeek.com/codigos-c/ejercicios-resueltos-de-c-3ra-parte/>
- [http://msdn.microsoft.com/es-es/library/2s05feca\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/2s05feca(v=vs.80).aspx)
- <http://msdn.microsoft.com/es-es/library/2s05feca.aspx>
- <http://casidiablo.net/matrices-en-c-sharp/>
- <http://www.latindevelopers.com/articulos/csharp/arreglos-en-csharp.php>. y ...
- Materiales compartidos en la página del moodle.



**¿Preguntas?**  
**¿Comentarios?**