

Arreglos en C#

Curso de Algoritmia y Programación

Prof. Luis E. Garreta U.
lgarreta@uao.edu.co

Universidad Autonoma de Occidente – Cali
Depto. Operaciones y Sistemas
Facultad de Ingeniería

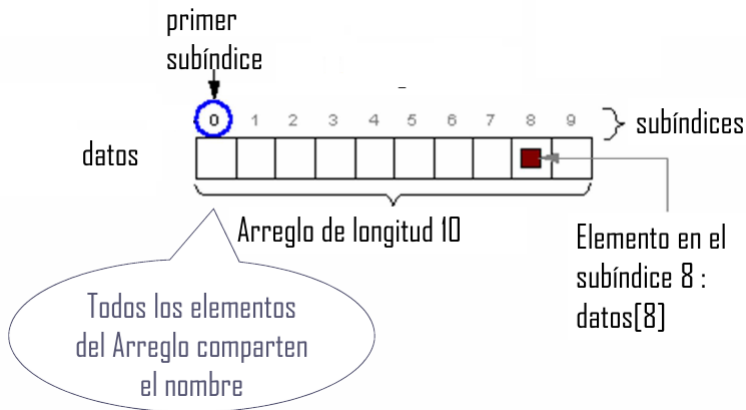
18 de octubre de 2018

Arreglos en C#

- ▶ En C# llamamos arreglo a un tipo especial de variable que puede contener una colección de objetos.
- ▶ Esta colección tiene longitud definida y no puede cambiar con el flujo del programa
- ▶ Esta limitación, permite un mejor desempeño que con cualquier otra colección.
- ▶ Usando el lenguaje podemos crear, recorrer y manipular arreglos de cualquier tipo de objeto.

Arreglos Unidimensionales

- ▶ Comenzando por lo más básico, los arreglos unidimensionales son colecciones lineales.
- ▶ Estas colecciones nos pueden ayudar a representar una secuencia numérica, letras del alfabeto, etc.



Sintaxis Definición de Arreglos

- ▶ Para crear un arreglo haremos uso de la palabra reservada `new` y los corchetes cuadrados `[]`.
- ▶ También es necesario conocer el tamaño que necesitaremos, ya que no es posible cambiar el tamaño una vez creado.
- ▶ **Sintaxis:**
`<tipo> [] <nombre> = new <tipo> [<tamaño>]`
`<tipo> [] <nombre> = new <tipo> [<tamaño>] {Elementos}`

Creación de Arreglos Vacios

```
char [] vocales = new char[5];  
  
int [] conteo = new int[10];  
  
object [] misObjetos = new object[3];
```

- ▶ Si creamos arreglos de esta manera, cada posición tendrá el valor por defecto del tipo de dato del arreglo:
 - ▶ El arreglo de vocales contendrá 5 `\0`, que es el valor por default de un `char`.
 - ▶ El arreglo conteo contendrá 10 `0`, que es el valor por default de un entero.
 - ▶ El arreglo misObjetos contendrá 3 `null`, que es el valor por default de un `object`.

Creación de Arreglos con Elementos

- Creación de un arreglo asignándole valores iniciales:

```
char [] vocales = new char[5] {'a', 'e', 'i', 'o',  
                                'u'};  
int [] conteo = new int[] {11, 22, 33, 44, 55};
```

- **new char[5]** { ..., nos permite indicar el tamaño y el tipo explícitamente y posteriormente indicar los valores, si cambiáramos el 5 por un 7 sin aumentar la cantidad de valores obtendríamos un error de compilación.
- **new int[]** { 11, 22, ..., nos permite indicar solamente el tipo de dato de los elementos, la cantidad es inferida por el compilador, en este caso podemos incrementar o reducir la cantidad de elementos al momento de inicializar sin ningún problema.

Acceso a los elementos de un arreglo

- ▶ Se utiliza el operador []
- ▶ La sintaxis es: <nombre arreglo> [<indice>]
- ▶ Ejemplo:

```
int [] conteo = new int[] {11, 22, 33, 44, 55};  
conteo [3]           // Accede a la posición 3 y recupera el valor 44  
int x = conteo [2]    // Guarda el valor 33 en la variable x  
int w = conteo [2] + conteo [3] // Se asigna a w la suma de 33 + 44  
int i = 4;  
int z = conteo [i]    // Utiliza una variable como indice
```

Modificación de los elementos de un arreglo

- ▶ Se utiliza el operador []
- ▶ La sintaxis es: <nombre arreglo> [<indice>] = <valor>
- ▶ Ejemplo:

```
int [] conteo = new int[] {11, 22, 33, 44, 55};  
  
conteo [2] = 66 // Cambia el valor de las posicion 2 por 66  
  
int i = 4;  
conteo [i] = conteo [2] + conteo [3] //Cambia el valor en posición 4 a 110
```


Recorrido de Arreglos

- ▶ Los arreglos pueden contener 1 o más elementos.
- ▶ Para procesar esos elementos utilizamos instrucciones repetitivas (for o while) para ir elemento por elemento

Ejemplo1: Imprimir todos los elementos de un arreglo

```
int [] conteo = new int[] {11, 22, 33, 44, 55};

// Imprimir todos los elementos del arreglo
for (int i=0; i < 5; i++)
    Console.WriteLine (conteo [i]);

// Calcular el cuadrado de cada valor
int i=0;
while ( i < 5) {
    int valor = conteo [i] * conteo [i]
    Console.WriteLine (valor);
    i++
}
```