ARREGLOS

PROGRAMACIÓN ORIENTADA A OBJETOS

INSTITUTO TECNOLÓGICO DEL SUR DE NAYARIT
MAESTRA: CINTHIA ANAHÍ MATA BRAVO

EDUARDO MARMOLEJO ORNELAS - 191140006

INDICE

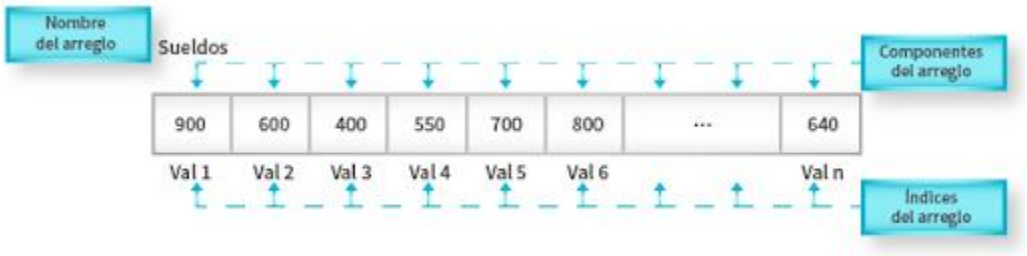
ARREGLOS

Un arreglo es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria RAM (sirve para almacenar datos en forma temporal).

Tipos de arreglos

Arreglos Unidimensionales

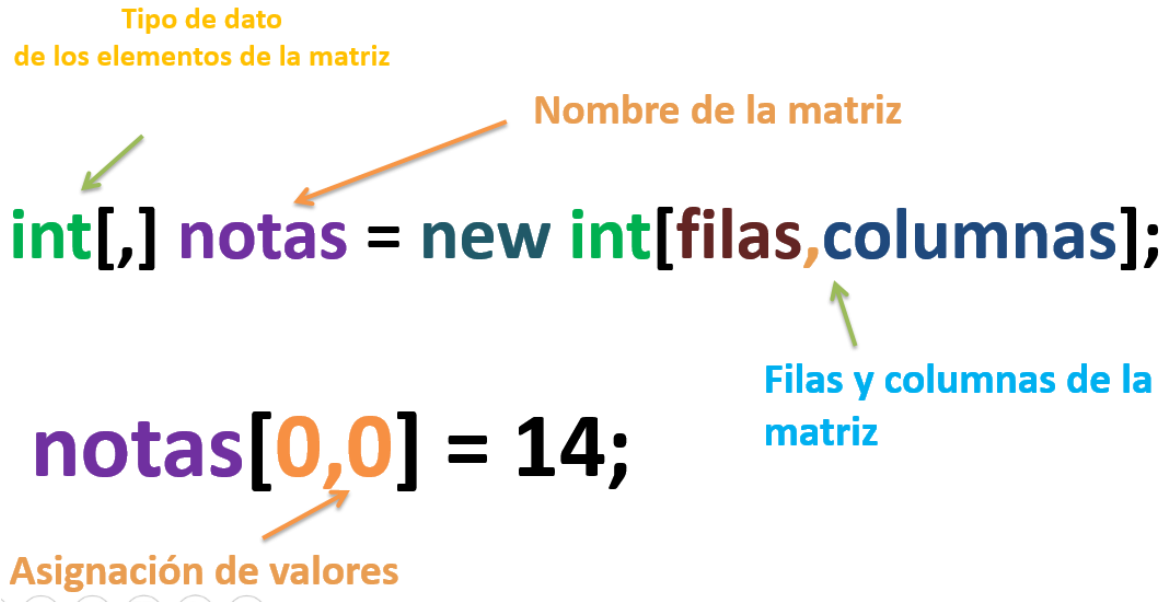
El arreglo unidimensional es un tipo de dato estructurado, formado por un conjunto de elementos del tipo de dato, almacenados bajo un mismo nombre y que se diferencian por la posición que tiene cada elemento dentro del arreglo.



Arreglos Multidimensional

Este es un tipo de dato estructurado y que este está compuesto por dimensiones.

Para hacer referencias para cada componente del arreglo se utilizan n índices (n=número de índices) para cada dimensión.



Declaración de los arreglos en C#

Los arreglos en C# (también conocidos como Arrays) al igual que en C/C++, son indexados iniciando en cero (0). La forma de trabajo es muy similar a la mayoría de lenguajes, pero hay lagunas diferencias que notarán.

Cuando vayamos a declarar un arreglo en debemos colocar los corchetes después de tipo de dato.

```
//Declaración erronea
   int valores[];


//Declaración valida
   int[] valores;
```

En C# podemos indicar el tamaño del arreglo luego de la declaración. Esto nos permite decidir qué tamaño va a tener el arreglo según lo que necesitemos.

```
int[] valores; //valores sin inicializar
valores = new int[100];  //100 elementos
valores = new int[20];  //ahora contiene 20 elementos
```

Arreglos Multidimensionales y Unidimensionales

En C# también podemos declarar arreglos multidimensionales, aquí unos ejemplos:

```
//Arreglos unidimensionales o de dimensión simple
int[] valores1;                          //sin inicializar
int[] valores2 = new int[50];

//Arreglos multidimensionales
int[,] valores1;                       //sin inicializar
int[,] valores2 = new int[3,7];
int[,,] valores3 = new int[3,4,2];    //Arreglo de tres dimensiones

//Arreglo de arreglos
int[][] matriz;                      //sin inicializar

//Los arreglos de arreglos se inicializan de manera diferente
int[][] matriz = new int[3][];
for (int i = 0; i < matriz.Length; i++)
{
   matriz[i] = new int[4];
}
```

Arreglos de Objetos de Clase

La función básica de un arreglo es almacenar en una variable más de un valor de un mismo tipo de dato, por ejemplo la siguiente declaración int[ ] numero= new int [5]; permite almacenar en la variable número, 5 valores enteros.

En las clases el concepto de arreglos es el mismo, con la diferencia que ahora se almacenarán objetos de una clase o de diferentes clases.

Los objetos se pueden estructurar como un array. Los objetos son variables y tienen las mismas capacidades y atributos que cualquier tipo de variables, por tanto, es posible disponer objetos en un array.

La sintaxis es exactamente igual a la utilizada para declarar y acceder al array. También disponemos de arrays bidimensionales.

Cuando se crea un array de objetos éstos se inicializan llamando al constructor sin argumentos. Por consiguiente, siempre que se prevea organizar los objetos en un array, la clase debe tener un constructor que pueda llamarse sin parámetros.

Declaración Arreglos de Objetos de clases en C#

Sintaxis para la definición del arreglo:

```
nombre_clase [ ] nombrevector = new nombre_clase[tamaño]; /*creación del espacio de memoria para el vector*/

nombrevector[x]= new clase( ); /*creación de las clases*/
```

Cuando necesitamos invocar algún elemento (propiedad o método) de la clase desde el

Programa principal lo hacemos así:

```
nombrevector[x].elementoinvocado; //si es una propiedad
nombrevector[x].elementoinvocado( ); //si es un método (si tiene parámetros no olvidarlos)
```

Recordemos que cada variación de x representa un nuevo objeto dentro del arreglo, con todos los atributos y métodos que implique.

Ejemplo de Uso de Arreglos en C#

```
//Declaración erronea
        int Cal[];
        //Declaración valida
        int[] Cal;

int[] Cal; //valores sin inicializar
        Cal = new int[10];  //100 elementos
        Cal = new int[20];  //ahora contiene 20 elementos
```

Arreglo Unidimensional

```
        int[] Cal1;                         //sin inicializar
        int[] Cal2 = new int[50];
```

Arreglo Multidimensional

```
        int[,] Cal1;                    //sin inicializar
        int[,] Cal2 = new int[3,7];
        int[,,] Cal3 = new int[3,4,2]; //Arreglo de tres dimensiones
```

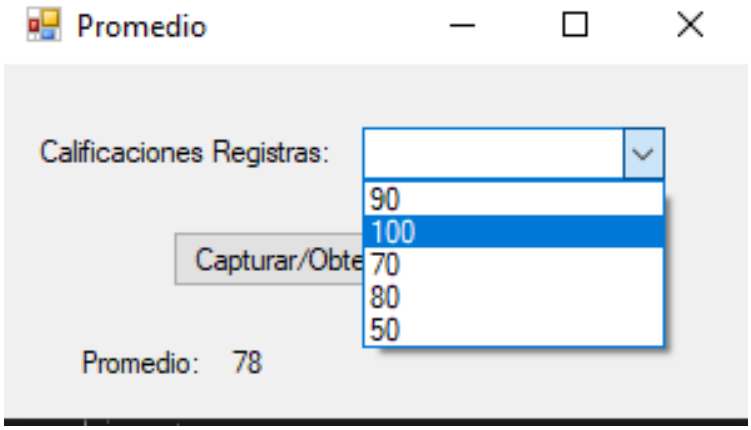
Conclusión

Al principio no le entendía, pero conforme investigaba le fui entendiendo, los arreglos son estructuras que almacenan datos, hay dos tipos de arreglos las unidimensionales que es un conjunto de elementos de cada tipo de dato y que están almacenados bajo el mismo nombre y las multidimensionales que están formadas por dimensiones y que utiliza n número de índices para cada dimensión.

En sí son fáciles y rápidos de usar, porque ya no se ocupará de programar una sola variable para que almacenen, y ya con estos métodos se realiza con facilidad.

Por ejemplo, después de esto se tenía que hacer un programa cualquiera pero que se utilizara este tema de arreglos, en mi caso tuve problemas porque no se podía hacer directamente de una clase, por cuestiones de codificación por parte del formulario, porque esta codificación de arreglos no se debe hacer en clases, esto se debe de hacer en el formulario, botón o donde se fuera a codificar, este otro dato que también se tiene que mostrar.

Para más sencillo los hice de manera unidimensional, ya que mi programa tiene que calcular el promedio de un estudiante e insertarlos en combobox, así como el de la imagen.



Y así como fue que hice el arreglo

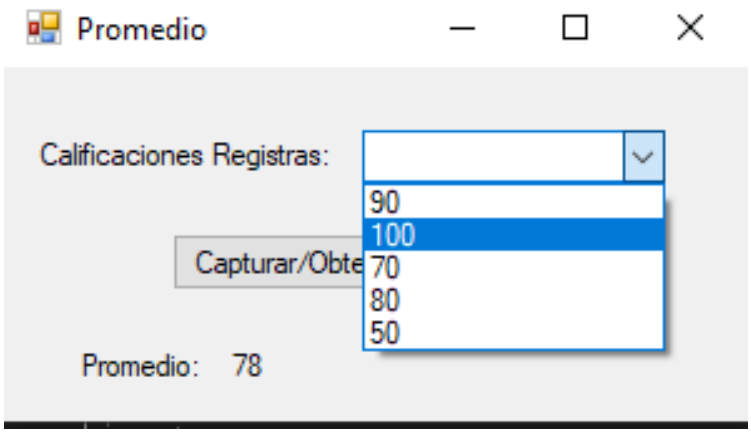
```
double[] Cal = new double[5];
Cal[0] = 90;
Cal[1] = 100;
Cal[2] = 70;
Cal[3] = 80;
Cal[4] = 50;
for (int x = 0; x <= 4; x++)
{
    cmbCal.Items.Add(Cal[x]);
    objPromedio.NumCal = objPromedio.NumCal + Cal[x];
}
objPromedio.Promedio();
lblPromedio.Text = objPromedio.Promedion.ToString();
```

También se utilizó lo que es la sentencia del for, para poder aplicar la suma y la división para sacar el promedio, hasta que hiciera de la primera imagen de esta conclusión.

Para este arreglo primero inicialice la variable en este caso Cal como double con 5 calificaciones, para esto se cuenta el cero como 1 y en como se ve la imagen, la quita calificación es el 4 y le puse predeterminadamente las calificaciones, aunque también se pueden agregar pidiéndolo desde el formulario y que aun así las capturara.

Aquí ocupe el for inicializando la x como el número de calificaciones, dándole la condición de que llegara a menor igual a 4 y que sumara 1 para cada calificación, después, y di valor al combo para que agregara las calificaciones previamente puestas

Hasta que el programa pudiera hacer esto.



Bibliografía

EcuRed. (2011). Arreglos (Informática). 2020, de Ecured Sitio web: https://www.ecured.cu/Arreglos_(Inform%C3%A1tica)

Ivan Cachicatari. (2017). Arreglos en C#. 2020, de Latindevelopers Sitio web: http://www.latindevelopers.com/articulo/arreglos-en-csharp/

UDB. (2019). Arreglos de Objetos en C#.. 2020, de UDB Sitio web: http://www.udb.edu.sv/udb/archivo/guia/informatica-ingenieria/programacion-ii/2016/i/guia-6.pdf