

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**HERRAMIENTAS DE
PROGRAMACIÓN –
C#**

Tema:

**PARÁMETROS DE MÉTODOS-PALABRAS
CLAVE REF, OUT**

Parámetros de Métodos en C#

En C#, los parámetros de métodos son los valores que se pasan a un método cuando se llama. Estos parámetros se definen en la firma del método y se utilizan dentro del cuerpo del método.

Los parámetros de métodos son los valores que se pasan a un método cuando se llama. Estos parámetros se definen en la firma del método y se utilizan dentro del cuerpo del método.

Palabra clave ref

La palabra clave **ref** se utiliza para pasar un parámetro por referencia. Esto significa que el valor del parámetro se pasa como una referencia a la variable original, en lugar de como una copia del valor.

La palabra clave **ref** se utiliza cuando se quiere pasar una variable como referencia, de modo que cualquier cambio en el valor del parámetro dentro del método afecte también la variable original fuera del método. Es decir, el parámetro debe estar inicializado antes de ser pasado al método.

Sintaxis y ejemplo

```

1  using System;
2
3  namespace ConsoleApp1
4  {
5      class Program
6      {
7          // Método que usa 'ref' para modificar el valor original
8          public static void Incrementar(ref int numero)
9          {
10             numero += 10;
11         }
12
13         static void Main(string[] args)
14         {
15             int valor = 5;
16             Console.WriteLine("Antes de llamar al método: " + valor); // Output: 5
17
18             Incrementar(ref valor);
19
20             Console.WriteLine("Después de llamar al método: " + valor); // Output: 15
21         }
22     }
23 }
24

```

En este ejemplo, el método **Incrementar** toma un parámetro **numero** con la palabra clave **ref**. Esto significa que el valor de la variable **valor** se pasa como una referencia a la variable original. Dentro del método, se incrementa el valor de **número**, lo que afecta directamente a la variable **valor** original.

Palabra clave out

La palabra clave **out** se utiliza para pasar un parámetro como un valor de salida. Esto significa que el método devuelve un valor a través del parámetro, en lugar de devolver un valor explícito.

La palabra clave **out** es similar a **ref**, pero se usa cuando el método debe devolver un valor a través de uno o más de sus parámetros. A diferencia de **ref**, no es necesario inicializar la variable antes de pasarla al método, pero dentro del método se **debe** asignar un valor al parámetro antes de que el método retorne.

Sintaxis y ejemplo:

```
1  using System;
2
3  namespace ConsoleApp1
4  {
5      class Program
6      {
7          // Método que usa 'out' para devolver múltiples valores
8          public static void ObtenerValores(out int num1, out int num2)
9          {
10             num1 = 10; // Se debe asignar un valor dentro del método
11             num2 = 20;
12         }
13
14
15         static void Main(string[] args)
16         {
17             int valor1, valor2;
18             ObtenerValores(out valor1, out valor2);
19
20             Console.WriteLine("Valor 1: " + valor1); // Output: 10
21             Console.WriteLine("Valor 2: " + valor2); // Output: 20
22         }
23     }
24 }
25
```

Diferencias entre ref y out

La principal diferencia entre **ref** y **out** es que **ref** requiere que el parámetro tenga un valor inicial antes de pasarla al método, mientras que **out** no requiere un valor inicial.

Además, **ref** se utiliza cuando se necesita modificar el valor original de la variable, mientras que **out** se utiliza cuando se necesita devolver un valor a través del parámetro.

Espero que esto te ayude a entender los parámetros de métodos en C# con las palabras clave **ref** y **out**. ¡Si tienes alguna pregunta o necesitas más ayuda, no dudes en preguntar!

Ejercicio práctico:

Vamos a crear un ejemplo que combine ambas palabras clave ref y out para devolver varios resultados y modificar un valor existente.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp1
8  {
9      internal class refOut
10     {
11         public void Calculo(ref int numero, out int cuadrado, out int cubo)
12         {
13             cuadrado = numero * numero;
14             cubo = numero * numero * numero;
15             numero += 10; // Modificamos el valor original
16         }
17     }
18 }
19

```

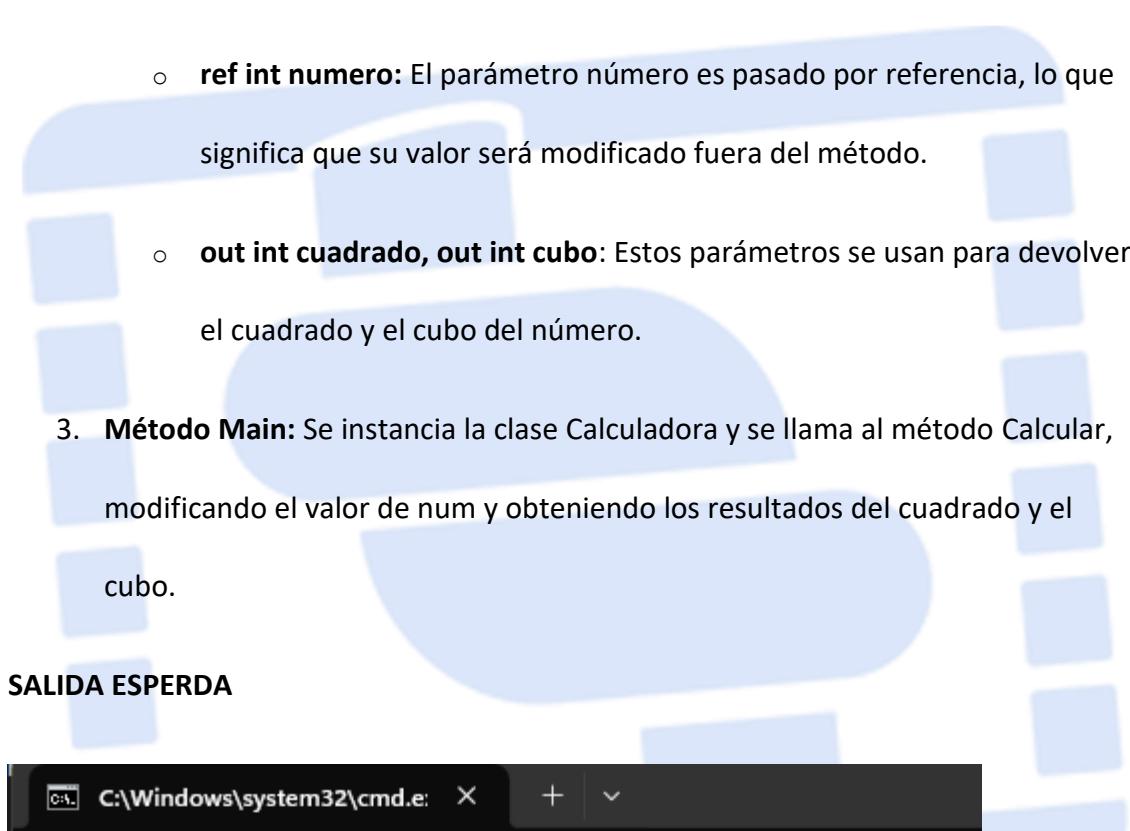
```

1  using System;
2
3  namespace ConsoleApp1
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int num = 2;
10             int cuadrado, cubo;
11
12             refOut cal = new refOut();
13
14             Console.WriteLine("Número original: " + num);
15
16             cal.Calculo(ref num, out cuadrado, out cubo);
17
18             Console.WriteLine("Número después de modificar con ref: " + num);
19             Console.WriteLine("Cuadrado: " + cuadrado);
20             Console.WriteLine("Cubo: " + cubo);
21         }
22     }
23 }
24

```

Explicación:

1. **Clase refOut:** Esta clase tiene un método público Calculo que toma el parámetro número por referencia con la palabra clave ref y usa out para devolver el cuadrado y el cubo del número.
2. **Método Calculo:**
 - **ref int numero:** El parámetro número es pasado por referencia, lo que significa que su valor será modificado fuera del método.
 - **out int cuadrado, out int cubo:** Estos parámetros se usan para devolver el cuadrado y el cubo del número.
3. **Método Main:** Se instancia la clase Calculadora y se llama al método Calcular, modificando el valor de num y obteniendo los resultados del cuadrado y el cubo.

SALIDA ESPERADA

```
C:\Windows\system32\cmd.e: + 
Número original: 2
Número después de modificar con ref: 12
Cuadrado: 4
Cubo: 8
Presione una tecla para continuar . . .
```

Este código muestra cómo se puede usar ref y out dentro de una clase, permitiendo que se modifiquen los valores originales y se devuelvan múltiples resultados a través de los parámetros out.

