#3 – Tipos de Datos

Antes de introducirnos en los tipos de datos de C# diremos que estos nos permiten indicarle a la computadora que tipo de información vamos a guardar en las variables declaradas, Es decir que al declarar una variable debemos además indicar de alguna manera qué vamos a guardar en ese espacio de memoria. No es lo mismo un caracter, que un párrafo, un número entero a uno con decimales. Recordemos que cuando declaramos una variable reservamos un espacio de la memoria de programa (RAM) para guardar un dato del tipo definido, para que se pueda determinar la cantidad real de memoria a reservarse se definen los tipos de datos.

Los tipos de datos y su tamaño se muestran en la siguiente tabla:

Como se declara en C#	Clase a la que pertenece	Tamaño	Qué podemos guardar
byte	Byte	8 bits, número entero sin signo	Números enteros dependiendo del tamaño permitido por el tipo
sbyte	Sbyte	8 bits, número entero con signo	
short	Int16	16 bits, número entero con signo	
int	Int32	32 bits, número entero con signo	
long	Int64	64 bits, número entero con signo	
float	Single	32 bits, número en punto flotante de simple precisión	Números reales dependiendo del tamaño permitido por el tipo
double	Double	64 bits, número en punto flotante de doble precisión	
decimal	Decimal	96 bits, número decimal	
bool	Boolean	8 bits, verdadero o falso	Verdadero o falso
char	Char	16 bits, caracter unicode	Un caracter
object	Object	Es la clase base para todos los tipos de datos	Cualquiera de los tipos
string	String	Cadena de caracteres, depende del tamaño de la memoria de programa (RAM)	Cadenas de caracteres unicode de tamaño fijo

A continuación se muestran algunos ejemplos

```
static void Main(string[] args)
        {
           //variable de la clase Byte, número entero sin signo de 8 bits
            byte valor = 233;
            Console.Write(valor);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Sbyte, número entero con signo de 8 bits
            sbyte valor = -120;
            Console.Write(valor);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Int16, número entero con signo de 16 bits
            short numero = 30000;
            Console.Write(numero);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable definida de la clase Int32, número entero de 32 bits
            int x = 20;
            Console.Write(x);

        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Int64, número entero con signo de 64 bits
            long numero = 1350000;
            Console.Write(numero);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Single, número en punto flotante de 32 bits
            float pi = 3.14f; //observar que se coloca el sufijo f o F
            Console.Write(pi);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Double, número en punto flotante de 64 bits
            double real = 25.789;
            Console.Write(real);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //variable de la clase Decimal, número en punto flotante de 96 bits
            decimal numero = 253.789999m; //Colocar sufijo m o M
            Console.Write(numero);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //Variable de la clase Boolean, valor lógico verdaredo o falso 8
bits
            bool flag = true;
            Console.Write(flag);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //Variable de la clase Char, un único caracter unicode 8 bits
            char letra = 'G'; //Observar que el dato se coloca entre comillas
            Console.Write(letra);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
         /*Variable de la clase String, texto unicode estático su tamaño
depende de la RAM*/
          string parrafo = "El siguiente es un párrafo de texto unicode estático";
          Console.Write(parrafo);
          Console.ReadKey();
        }
```

Como se puede apreciar, no se ha ejemplificado al tipo object, ya que el mismo tendrá un tratamiento especial en otro apunte.

Es importante comentar también que hay una palabra clave en C# que nos permite declarar variables con asignación implícita de tipos, dicha palabra es var. La variable declarada

tomará el tipo a partir de la expresión de la derecha de la instrucción de inicialización, por ejemplo:

```
static void Main(string[] args)
        {
            //Asignación implícita mediante la palabra reservada var
            var parrafo = "la variable párrafo tomara el tipo string";
            Console.Write(parrafo);
            Console.ReadKey();
        }
```

```
static void Main(string[] args)
        {
            //la variable valor tomará el tipo int
            var valor = 33;
            Console.Write(valor);
            Console.ReadKey();
        }
```