

## TIPOS

Los tipos en C# se clasifican en: tipos *valor* y tipos *referencia*. Una variable de un tipo *valor* almacena directamente un valor (datos en general), mientras que una variable de un tipo *referencia* lo que permite almacenar es una referencia a un objeto (posición de memoria donde está el objeto). Por ejemplo:

```
int suma = 0;           // suma almacena un entero.
string cadena = null; /* cadena permitirá almacenar una
                       referencia a un objeto string. */
```

La tabla siguiente resume los tipos intrínsecos en C#.

Tipo C#	Estructura C#	Bytes	Rango de valores
<b>bool</b>	Boolean	?	<b>true y false</b>
<b>byte</b>	Byte	1	0 a 255
<b>char</b>	Char	2	0 a 65535 (U+0000 a U+ffff)
<b>DateTime</b>	DateTime	8	1/Enero/1 a 31/Diciembre/9999 00:00:00 AM a 11:59:59 PM
<b>decimal</b>	Decimal	16	+/-79228162514264337593543950335 ó +/-7.9228162514264337593543950335E+28
<b>double</b>	Double	8	+/-1.79769313486231570E+308
<b>int</b>	Int32	4	-2147483648 a +2147483647
<b>long</b>	Int64	8	-9223372036854775808 a +9223372036854775807
<b>object</b>	Object	4	Cualquier tipo puede ser almacenado en una variable de tipo <b>Object</b>
<b>sbyte</b>	SByte	1	-128 a 127
<b>short</b>	Int16	2	-32768 a 32767
<b>float</b>	Single	4	+/-3.4028235E+38
<b>string</b>	String	?	0 a 2 billones de caracteres UNICODE
<b>uint</b>	UInt32	4	0 a 4294967295
<b>ulong</b>	UInt64	8	0 a 18446744073709551615
<b>ushort</b>	UInt16	2	0 a 65535
<i>Estructuras</i>			Tipos definidos por el usuario

(? = depende de la plataforma de desarrollo)

Todos los tipos primitivos expuestos tienen una estructura de datos asociada; por ejemplo, el tipo **double** es un alias de **System.Double** (estructura **Double** del espacio de nombres **System**), **char** es un alias de **System.Char**, **bool** es un alias de **System.Boolean**, **int** es un alias de **System.Int32**, etc. Por lo tanto, un dato de un tipo primitivo es un objeto.