

El sistema de tipos de C# está unificado, de tal forma que un valor de cualquier tipo puede tratarse como un `object`. Todos los tipos de C# directa o indirectamente se derivan del tipo de clase `object`, y `object` es la clase base definitiva de todos los tipos. Los valores de tipos de referencia se tratan como objetos mediante la visualización de los valores como tipo `object`. Los valores de tipos de valor se tratan como objetos mediante la realización de *operaciones de conversión boxing* y *operaciones de conversión unboxing*. En el ejemplo siguiente, un valor `int` se convierte en `object` y vuelve a `int`.

```
using System;
class BoxingExample
{
    static void Main()
    {
        int i = 123;
        object o = i;    // Boxing
        int j = (int)o;  // Unboxing
    }
}
```

Cuando se convierte un valor de un tipo de valor al tipo `object`, se asigna una instancia `object`, también denominada "box", para contener el valor, y el valor se copia en dicho box. Por el contrario, cuando se convierte una referencia `object` en un tipo de valor, se comprueba si la referencia `object` es un box del tipo de valor correcto y, si la comprobación es correcta, se copia el valor del box.

El sistema de tipos unificado de C# conlleva efectivamente que los tipos de valor pueden convertirse en objetos "a petición". Debido a la unificación, las bibliotecas de uso general que utilizan el tipo `object` pueden usarse con tipos de referencia y tipos de valor.